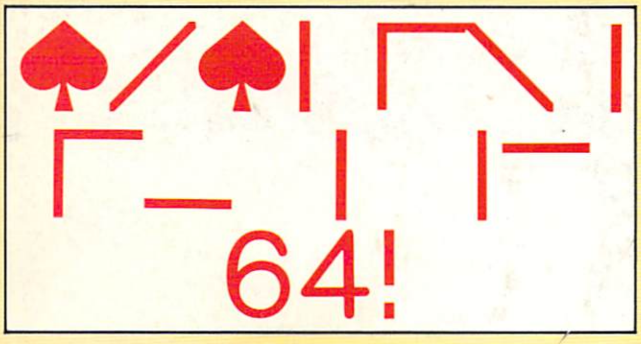
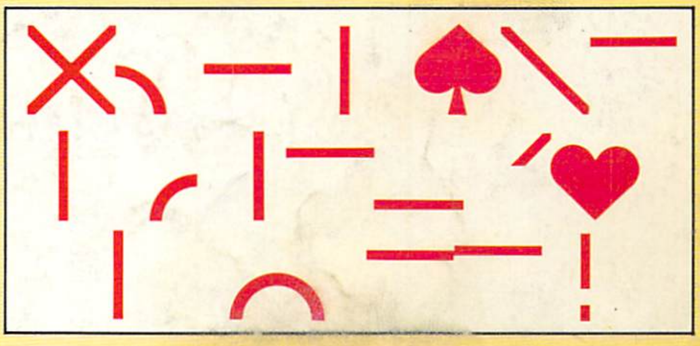
48355

Ahoy! COMMODORE USERS

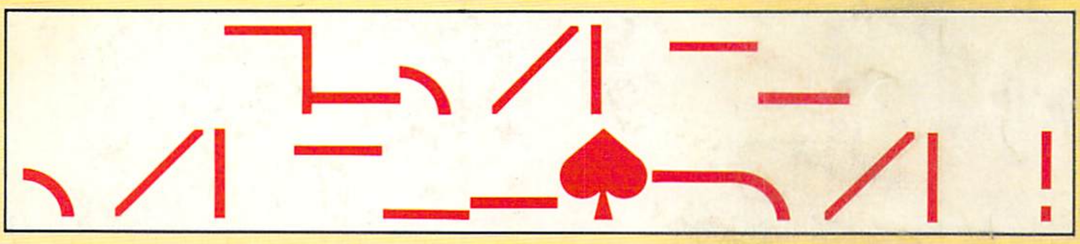
K

Ion International, Inc. $2.50/Can. $2.75 March 1984

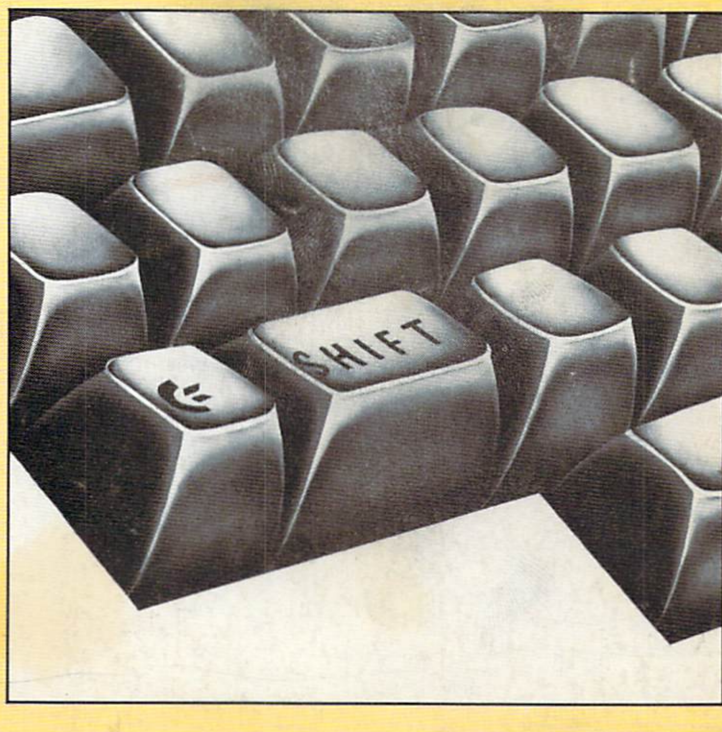




PLUS!



CAN YOU READ THIS? IF NOT, PRESS THESE KEYS & TURN THE PAGE!

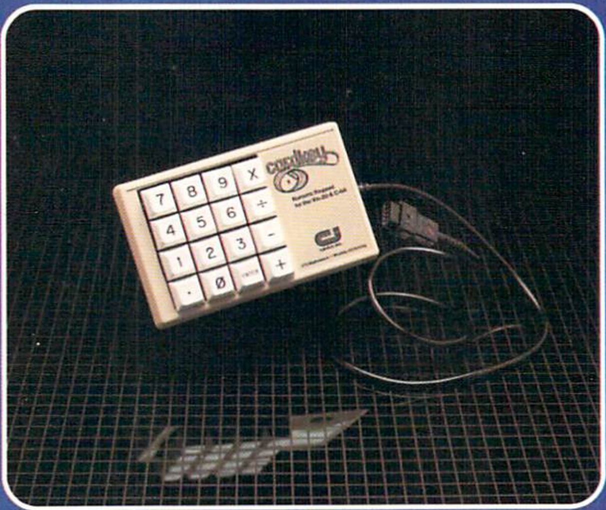


0 71896 48355 03

Cardco Presents

16 Key Keypad for the VIC-20™ and Commodore 64™

The CARDKEY/1 provides an additional programable 16 keys to either the VIC or 64. This is a separate pad that plugs into the joystick port of your computer. It comes with tape programs (transferable to disk) that allow the user to define the 16 keys as any number, letter or character string value. It additionally allows the keys to be defined as strings. Another function available is the direct calculator mode in which you use your computer as a calculator. For example you press 5 + 5 (enter) and the computer prints 10 on the screen.



Manufacturer's Suggested Price: $39.95

cardkey 1

The CARDKEY features:

- 16 keys laid out in a calculator format
- All keys are user definable
- Keys can be defined as any character
- Keys can be defined as complete strings
- Optional calculator mode



See a complete line of American made Cardco Products at a computer store near you, today.

313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

cardco, inc.

Reader Service No. 90

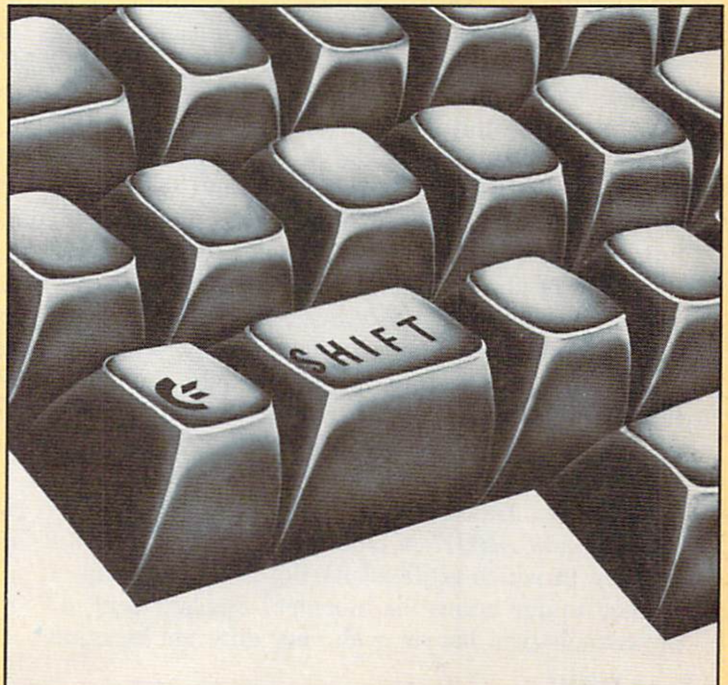
48335

$2.50/ $2.75 1984

ANATOMY OF THE 64!

VIC GAME BUYER'S GUIDE!

PRINTER INTERFACING!



WHY STOP NOW? KEEP TURNING!

VIEW FROM THE BRIDGE

Ahoy, mateys! Whether you're climbing on board for the first, second or third time, we welcome you to the fastest-rising Commodore magazine this side of Davy Jones' locker! (And, no, we're not going to keep up this ridiculous nautical jargon for the entire page.)

The response to our first two issues has been encouraging. Judging from the comments of those of you who've written in and those we met at January's Consumer Electronics Show, we're well on our way to producing the kind of magazine you've been waiting for. Keep those comments, both good and bad, (one more nautical pun) flooding in!

And we'll keep improvements flooding your way. Last issue, we added more color. This issue, we added more artwork. Next issue, we're adding more pages, and plan to do the same each issue for several months to come.

But as for this issue:

Last time, Morton Kevelson took you on a *Fantastic Voyage*-type tour of the 1541 disk drive. This issue, Richard Herring reveals to you *The Anatomy of a C-64.* (Turn to page 21.)

But Richard hasn't spent his whole life taking apart computers. A certified teacher who spent several years in the Florida elementary school system and an educational software consultant, he's eminently qualified to undertake the multipart series on choosing educational software that begins in this issue. So many educational programs wither and die on your children's shelves, or if used, don't do the job. You'll save many dollars you might otherwise waste by reading *Educational Software: A Guide for Parents.* (Turn to page 41.)

We've kept Morton K. busy, too. The first half of his definitive piece on the ins and outs and hows and whys of *Printer Interfacing for the Commodore 64 and the VIC-20* will dispel your doubts on this little-explored subject. (Turn to page 27.)

But we're not neglecting you VIC owners. with many games for your system starting to appear at marked-down prices, you'll want to know which ones to take a close look at. Walter Salm goes over the best bets in the entire run of VIC game software in his *VIC Game Buyer's Guide.* (Turn to page 49.)

Remember David Ritchie's *Golden Gateway* in the February *AHOY!:* a chilling, thrilling vision of a future in which artificial intelligence will be implanted in our brains via biochips? Stephen Bent, a biotechnological lawyer with very different ideas on mankind's future, rebuts David in *Beyond the Golden Gateway.* (Turn to page 31.)

Dale Rupert's instructional ideals remain as lofty as ever, as he leads you into *The City of Truth* for a look at logical programming. (Turn to page 17.)

The team of Michael Kleinert and David Barron conclude their three-part series on file management this issue with an explanation of *Using Random Files on the Commodore 64.* (Turn to page 37.)

With all three methods of file management under your programming belt, you'll want to start throwing some weight around at the keyboard. So study the same team's *Dynamic Power for your 64!* to learn how to give your computer an exciting new capability. (Turn to page 47.)

Game wizard Robert Alonso has returned, with *Space Lanes* for both the 64 and the VIC. (Turn to page 34.)

Typing in all those programs should keep you far too busy to write letters or make phone calls. But if you punch in one additional program—B.W. Behling's *Easy Access Address Book*—you'll at least be able to find the addresses and phone numbers you need more easily than ever before. (Turn to page 39.)

Commodares, back by popular demand, provides more puzzles for any and all readers willing to accept the challenge. (Turn to page 45.)

Of course, in *Scutttlebutt* you'll find the latest happenings in the Commodore-conscious world, including the new Commodore 264 and 364 computers and many other products just announced at the January Consumer Electronics Show. (Turn to page 10.) In our *Reviews* section you'll find detailed examinations of some of the worthiest (and least worthy) new releases for both the C-64 and the VIC. (Turn to page 57.)

Between the features described and the other surprises you'll find inside, we're proud of the way we've packed this issue of *AHOY!* with material of concern to you, the Commodore user looking to expand his programming knowledge in new and diverse directions. And if there's one promise we feel qualified to make, given the unpredictable nature of the industry we're in, it's that we're only going to get better as we get bigger.

We hope to meet with you in these pages each and every month. If you can find time to drop us a line as well, we'd like that even better.

But don't let us hold you up any longer. Sail right into the third, best-ever issue of *AHOY!*

—Steve Springer

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Editor
Steve Springer

Managing Editor
Robert J. Sodaro

Senior Editor
Tim Moriarty

Consulting Editors
Ben Bova
Morton Kevelson
Dale Rupert

Art Director
Joan Mazzeo-Weinshank

Art Production
Eve Griffin
Peter Neiburg
Raoul Tenazas

Publisher
Michael Schneider

Advertising Representative
JE Publishers Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213)467-2266
Boston (617)437-7628
Dallas (214)660-2253
New York (212)724-7767
Chicago (312)445-2489
Denver (303)595-4331
San Francisco (415)864-3252

AHOY! No. 3, March 1984. Published monthly by Ion International Inc., 45 W. 34th St., Suite 407, New York, NY, 10001. Subscription rate: 12 issues for $19.95, 24 issues for $37.95 (Canada $23.95 and $45.95 respectively). Application to mail at 2nd Class postage rates is pending at New York, NY and additional mailing offices.

All editorial and subscription inquiries and software and hardware to be reviewed should be sent to 45 W. 34th St., Suite 407, New York, NY 10001.

SYNAPSE EXCITEMENT

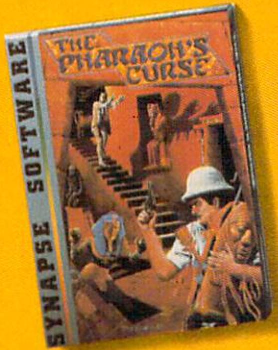


On patrol
Out of the sun comes your RAF biplane, loaded down with a deadly cargo of bombs and bullets. But watch out for the anti-aircraft guns and the enemy fighters—a hit could mean a tricky landing for repairs and ammo. **BLUE MAX.***





Ancient treasure
A fortune is yours for the taking. But can you avoid the ghost of Rama and the evil mummy? Are you nimble enough to leap the chasms and outsmart the booby traps between you and freedom? **The PHARAOH'S CURSE.***





Spellbinding
Only you can restore the forest through ancient spells. Then you must march your army of enchanted trees into battle against the Troglodytes and the evil Necromancer. Who will emerge triumphant from the final conflict? **NECROMANCER.***





Take the controls
Your helicopter mission—capture vital fuel and weapons, free the enslaved masses, and finally destroy the fortress itself. Will you triumph or be crushed by the fiendish Kraalthan lords? **FORT APOCALYPSE.***



EDITORIAL

By Raymond Z. Gallun

Back at the turn of the twentieth century, L. Frank Baum's *Wonderful Wizard of Oz* first appeared—a charming, whimsical fantasy. In it, the Tin Woodman had once been a flesh-and-blood man; but through wicked witch-directed self-maimings with his axe, he lost his legs, arms, head, body, and heart. Except for his heart, all these parts were successfully replaced by a friendly tinsmith, until the Woodman became entirely a metal robot, though without activating mechanisms of any sort; apparently all his functions were driven by pure magic.

In this writing, L. Frank Baum was suggesting the oldest of human yearnings, seldom written about by anybody before: betterment, healing, widened range and ruggedness, freedom, conquest of pain and limitation, broadened comprehension of, and contact with, the universe.

By the early 1930s, such visions had moved somewhat beyond the impossible-dream phase, and there were glimmerings of how such wistfully viewed phenomena might be accomplished on a real, scientific basis. At least this seemed so in the then-expanding field of science fiction.

I don't know whether I thought consciously of Baum's Tin Woodman or not, but in January, 1935, a little story of mine, *Mind Over Matter,* was published in *Astounding Stories.* In it, a test pilot suffers a misfortune similar to the Tin Woodman's; he crashes in a new aircraft, and nothing of his body is salvageable except his brain. The difficulties in giving him a mechanical body are enormous, but mind-supported medical technology of the future time of the story is equal to this matter problem. The tiny electrical impulses meant to be transmitted from his brain to his living flesh along motor-nerve channels are electronically interpreted, so that he has complete control of muscles of literal steel. Similary, by a reverse process, he has sensory perception from artificial eyes, ears, fingertips. He has become a mighty machine of metal and plastics, roughly manlike in shape. Being nuclear-powered, he has no need for food or oxygen to breathe, beyond the rather small requirements of the single, alive, and well insulated part of him that houses his mind and feelings. He is impervious to cold, and to anything less than red-heat. He is vastly stronger and more rugged than he used to be. He can survive comfortably in almost any environment.

But his great horror is his loss of the capacity to love and be loved in a human manner. He is enraged to the point of murder with his scientist-physician friend who saved him for this sorry state. So regretfully, his friend introduces ''adrenalin and something else''—presumably mood-changing hormones—into the blood-system that nourishes his brain. Again mind triumphs over matter. His attitude does a turn-about. He begins to feel euphoric about his new advantages. And his friend reassures him more substantively: his human limitations have dwindled. He is now almost immortal. Many strange and wondrous experiences lie ahead. And with time and further progression, means may be found to give him a living human form again—if he wants it. So he looks up at the full moon, and it ''beckons like the call of home.'' He is ready for space, other worlds, maybe even other star-systems.

This little story, almost forgotten, seems to be surfacing again in the perceptions of some people. Is it because parts of it are coming true?

For instance, an artificial heart, practical enough to be fuss-free most of the time, seems close to reality, promising to end one more human hazard and limitation. Also the ionic nature of nerve-signals, both motor and sensory, are becoming better understood and applied. Prosthetics, in which such brain-messages can be used to guide the movements of artificial limbs and fingers, are already in the developmental stage. Soon there may be mechanical hands as effective as living ones, and cosmetically indistinguishable.

There is a unity and oneness in all advances in know-how. We are already in space, an ultimate frontier, where the distance we can go, and the things we can learn about are as yet unmeasured.

Living brains are one thing, but our present computers are parallel in function, and far faster. There are memory banks. The logic-process, that portion of intellect that researches knowledge in the direction of solving posed problems toward the best possible solutions, and forms new associations of data, thought and invention, is pretty well understood, and has been in increasingly broad and keen computer-use for years: chains of simple questions, answerable by yes or no, go or no-go, zero or one, leading to complex answers. Electrons finding their way along the best, most logical paths, are quick, but photons will be quicker, more selective and efficient. Computers with such forms of inventive intelligence seem a definite prospect.

But consciousness—awareness of individual self...can we even define it fully? I can't, though

I've tried. Lacking even a clear definition, how can we ever give it to a device? As they are today, our most advanced computers are probably no more aware than the simplest mechanical adding-machine, summing two and two. They don't feel; they can't become bored or frightened or angry, or truly eager for some hoped-for objective; they don't need will or enthusiasm to do their work; they are driven only by how they have been programmed, and by the energy-flow within them. In contrast, it is our human emotions that propel our wits into action—fascination, curiosity, ambition, fury, altruism, need. All of these seem functions of our consciousness.

Yet who can say? The door of possibility is open wider than ever before. Nimbler artificial intelligences still in the visionary stage may yet be programmed to help give later generations of themselves the mystery of being aware and interested, maybe by some interfacing of physics with biology since, to this date, consciousness seems strictly a biological phemomenon, in people and in at least the higher beasts. This is another might-be that can frighten or thrill us, rightly or wrongly.

Many things might soon come about: memory- and wisdom-enhancers implanted against our skulls? Or devices giving us new senses, enabling us to communicate indivdually with each other, even over great distance, still by electron-magnetic means, but without the slanted intervention of TV or radio? A force for improved understanding and warmth, worldwide? Which of our already-born children or grandchildren will be living and working on the moon, or Mars, or inside some luxurious, hi-tech space-construct? Will a life-span go to two centuries? . . .

Or will few of the good things get a chance to happen, in the face of other aspects of our hurtling technology, as some persons predict?

We live in a wondrous scary time of rapid progress and widened boundaries. I hope we have what it takes to adjust to our situation. Whatever our individual views, we're stuck with it. We've already come a long way—reaped many benefits—without ultimate disaster. I'm an optimist. There are signs that speed of computer reaction-time, in combination with other advances, may soon neutralize the threat of nuclear war that has troubled us for so long.

Of all the historical eras so far, I wouldn't choose any other. The universe is before us.

Raymond Gallun's first published story appeared in 1929 in Air Wonder Stories. *Since that time he has authored dozens of short stories and novels on a multitude of subjects in the science fiction field. After a long retirement, he recently returned to the field with two novels.*

SCUTTLEBUTT

WINTER '84 CONSUMER ELECTRONICS SHOW • COMMODORE 264 AND 364 • MAYFLOWER VOYAGE SIMULATION • STOCK MARKET INSTRUCTION • MUSIC SYNTHESIZERS • SECURITY SYSTEM • JEWISH SOFTWARE • BUSINESS AND EDUCATION PROGRAMS • ON-LINE SOFTWARE LIBRARY • CHEATSHEETS • BLACKJACK TEACHER • NEW GAMES UPDATE



Koala Pad ***helps bare your talents.***
READER SERVICE NO. 65

CAVE ART

While not brand new, the *Koala Pad* from Koala Technologies Corporation has apparently escaped the notice of many C-64 and VIC-20 users. We've been having a lot of fun with it, and wanted to let those of you who've been hibernating in on it.

After loading the supplied software, you can create onscreen designs by drawing freehand on the pad or choosing from the provided shapes. What you draw can be reproduced in mirror image to create symmetrical patterns, in up to 16 different colors. *Coloring Series I* adds 26 geometric designs that can be combined into countless shapes. *Illustrator* lets you render three-dimensional graphics like those pictured. Specially designed games are also available, including *Spider Eater* (protect the treble clef of an onscreen keyboard from invading arachnids while playing songs) and *Spellicopter* (pilot a chopper through dangerous skies to recover secret words).

To locate the dealer nearest you, call (800)227-6703; in California, (800)632-7979.

88's ON THE 64

Music synthesis programs come and go with the speed of a string of hemidemisemiquavers. Still, two more manufacturers are shooting for a place in the Top 40 with new lines of musical software.

MusiCalc 1 (Synthesizer and Sequencer) from Waveform turns your C-64 into a three-voice synthesizer with interactive real-time sequencing, slide controls, modulators, and transposers. You select elements like tempo, key, and musical style from a colorful display, then play along with preprogrammed melodies or create and store your own in styles ranging from classical to new wave to Japanese. While complex enough for the expert, *MusiCalc 1* requires no musical experience; in fact, the program lists the notes and rests you select in a form of notation easily understood by musical illiterates. For the purist, *MusiCalc 2 (ScoreWriter)* will translate your original compositions into standard musical notation. *MusiCalc 3 (Keyboard Maker)* includes over 30 preset scales from around the world, and allows you to create your own custom keyboards. Both require the use of *MusiCalc 1*, as do *MusiCalc Templates 1 (African and Latin Rhythms)* and *2 (New Wave and Rock)*, companion disks providing additional scores and patches.

You'll have to pay the piper $74.95 for *MusiCalc 1; 2* and *3*, $34.95 each; *Templates 1* and *2*, $29.95 each. The *MusiCalc Professional System* offers all five pieces for $174.95. Demo disk is $5.00.

From Entech comes *Studio 64*, described as a "music word processor." The notes you play appear on the screen, and scroll across it for easy reading. With *Add Mus'in* you can add your compositions to other programs. (How about *Color My World* with your graphics program? *Call Me* with your telephone file? *Folsom Prison Blues* with your tax spreadsheet?) These two, plus the Lead Sheet Writer, sell for $39.95 each on disk or cassette.

Waveform Corporation, 1912 Bonita Way, Berkeley, CA 94704. Phone: (415)841-9866.

Entech, 8224 Sunland Boulevard, Sun Valley, CA 91352. Phone: (213)768-6646.



MusiCalc ***lets amateurs compose.***
READER SERVICE NO. 66



Commodore 264: next generation.
READER SERVICE NO. 67

VICTORY AT CES

It would have been difficult to attend the 1984 Consumer Electronics Show (held January 7-10) without gaining a clear insight into which home computer has become the industry leader. As editors Tim Moriarty and Steve Springer walked up and down the aisles at the Las Vegas Convention Center, it was rare to find a software manufacturer whose list of compatible systems did not include the Commodore 64.

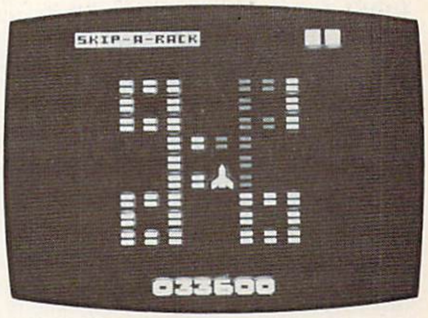
Between innumerable third party releases and a surprisingly bountiful crop from Commodore itself, 64 and VIC users have a shipload of software and hardware coming their way—so much that we can only begin to report on it in this issue. Those of you who remember the dearth of available Commodore software that existed at this time last year will greet that as especially good news.

But did CES bring bad news for Commodore users as well? What was that you read in the paper about two new home computers—the Commodore 264 and the Commodore 364—with built-in software, reduced size and weight, and increased capability? Has your 64 become a dinosaur? Has your VIC become a fossil?

There's good news here, too—the 264 and 364 are not, to say the least, a quantum leap beyond the 64 in technology. Not a leap that would justify the wait between the time you bought your 64 and late spring of this year, when the new computers become available—or the higher price tag. More about that below.

But first, the latest products by Commodore of interest to C-64 and VIC-20 users.

- The *Magic Voice* speech module, with a built-in vocabulary of 235 words, letting the user program directly from BASIC and/or assembler. Music, graphics, and speech can be programmed simultaneously. In addition to a port for other talking and non-talking cartridges (up to 128K), the module supports an audio out for linkup with a hi-fi, TV, or monitor.
- Talking games for use with *Magic Voice,* including the educational offerings *A Bee C's* and *Counting Bee* from Commodore, and Bally Midway adaptations of arcade games *Gorf and Wizard of Wor.*
- *3-Plus-1,* a single program containing a word processor, electronic spreadsheet, file manager, and business graphics package that can share and trade information back and forth. It includes windowing capabilities, letting the user view the word processor and spreadsheet simultaneously. Available on cartridge for the 64 and 264 (also as a built-in option for the latter).
- *Commodore LOGO,* featuring 170K available disk storage, music capability, picture saving, seven programmable sprites, assembly language interface, arithmetic floating decimal point, program tracing, comment/remark capability, word and list commands, and graphics and word and list command tutorials.
- 64 and 264 adaptations of Island Graphics' *Micro Illustrator,* allowing computer artists to draw freehand or generate geometric figures using a variety of brushes, colors, and textures, with either a joystick or a lightpen. The 64 version is scheduled for February release; the 264, for April.
- *Magic Desk II,* allowing access to the integrated text editor, spreadsheet, file manager, and calculator through onscreen pictures instead of commands.
- *Superscript 264,* a word processor with text editing, number calculations, mail list functions, and a cut and paste feature.
- *Easycalc 64* and *Easycalc 264,* cartridge-based spreadsheet programs featuring color selection and graphics.
- *B/Graph,* a business graphics and statistics package that produces 3D charts, graphs, pie charts, histograms, and other graphics.
- *Financial Advisor,* a cartridge-based aid in computing loan, mortgage, and investment formulas.
- *Teligraphics,* videotext and graphics software allowing modem-transmission of pictures, text, and business graphics.
- The first four in the new Gold Medallion series of adventure games: *International Soccer* (realistic 3D action), *Viduzzles* (kids' video puzzles), *Jack Attack*



Solar Fox ***(CBS version shown).***
READER SERVICE NO. 68

(animated strategy contest) and *Solar Fox* (adapted from the Bally Midway hit).

• Ten new educational programs, including *Math Facts, Numbers Galore, Frenzy/Flip Flop, Gulp!/Arrow Graphics, Easy Count/Easy Match, What's Next/ Letters or Numbers, A Letter Match/More or Less, Shapes & Patterns/Group It, Chopper Match,* and *Type Right.*

• The first in a series of applications templates for the *Manager 64* database system, consisting of *Home Manager, Kitchen Manager, Sports Manager,* and *Business Manager.*

CHIPS ON THE HORIZON

To everyone in the world save our readership of happy 64 and VIC owners, the most intriguing Commodore announcement to come out of CES was that of the new 264 and 364 computers. Just this once, we'll take the novelist's, not the journalist's, approach—we'll save the most exciting stuff for last.

The 264 and 364 will have no sprites. Bit map graphics and text graphics are virtually the same, with one plus: in addition to 16 colors, there are 8 luminescence values. (When Commodore talks about the unit's "128 colors," they mean 8 X 16.) Any character can be designated a flashing character.

While the C-64 has 16K of ROM for the operating system and BASIC (20 including the character ROM), the new units will have 32K. This means extended commands, support of graphics commands with bit map commands, sound commands, and functions to read joystick commands. The BASIC will also have structured programming commands, and editing commands such as autoline numbering/renumbering and delete.

There will also be a built-in ML monitor, larger than that of the PETs or the 8032, because it will include assemble/disassemble commands.

And finally, the main distinguishing feature between the 264/364 and the 64: built-in applications software. Though the titles have not yet been specified, they will include word processing, spreadsheet, and business graphing. When you buy your machine, you choose the software you want. The 264 can have one built in; the 364, two. (That's one of the two main differences between the 264 and 364, the other being built-in speech synthesis on the higher-end model.) Whatever applications you must forego at the time of purchase can be added on cartridge.



Former CEO Jack Tramiel remains as a consultant to Commodore.

As for compatibility, the 1541 disk drive will work with the new computers. The 801 printer should work. The cassette drive will not. A new, fast-parallel disk drive will be introduced.

All of the above courtesy of Sheldon Leemon, freelance expert on all things Commodore whose indepth analysis of the 264/364 will appear in an upcoming issue.

GOODBYE, MR. CHIPS

"If you came to hear bad news," said one of the speakers at Commodore's CES press conference, "you came to the wrong place." That's because Commodore waited several more days to drop its bombshell.

Jack Tramiel, Commodore's founder, president, and chief executive officer, resigned on January 13, less than a week after standing at the same dais and basking in the glory of building Commodore into the world's number one microcomputer manufacturer.

Mr. Tramiel claimed to have been toying with the idea of retirement for three years. Now, with Commodore a $1 billion a year business, he felt the time was right to leave. "I'm not a professional executive," he said. "I use a personal touch like a family business. I never believed I would be able to run a business of that size." *The Wall Street Journal,* however, speculated that the resignation stemmed from disagreements between Trameil and Irving Gould (chairman of the board and chief stockholder) over the appointment of a new executive to head Commodore's worldwide operations.

Whatever the reason, Tramiel's Friday resignation dropped the value of Commodore's stock from $47.75 to $45.75 on Monday. Investors know it was his aggressive pricing and marketing policies that drove the 64 and VIC first down in price, then to the top of the home computer heap. They know that his will be

a tough act to follow.

BUSINESS

For those who toe the bottom line, a brief rundown of Commodorian business news.

• *Management System 64* is a C-64 software package for the small businessman from En-Tech Software.

System 64 records sales transactions, prints up invoices and packing slips and stores that information. It generates sales reports and customer lists, and prints out mailing labels of all clients. The program also keeps track of inventory, the information being entered as it is received and as it rolls out. $79.95.

En-Tech Software, P.O. Box 881, Sun Valley, CA 91353.

• *Trendline,* for the 64 from Image Computer Products, is a stock market program. It stores daily closing data and calculates slopes, means, and standard deviation over any portion of the data. It will store up to 400 inputs (more than a year's worth of daily activity data). The program was designed by a stock market analyst.

Image Computer Products, P.O. Box 3761, Cherry Hill, NJ 08034.

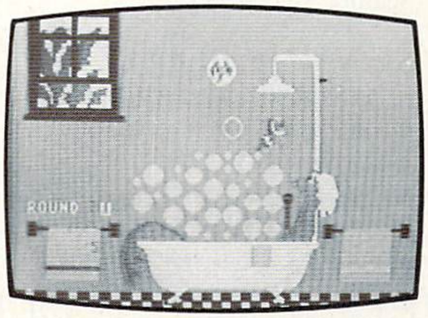
• Insta Software is making many of its 64 business/management programs available in two sets of integrated packages.

The *Investment Combo* program integrates, under a master menu, these three programs, which are also available separately: *Insta-Calc* (financial spreadsheet in a grid format), *Insta-Graph* (bar and line graphs creator/editor. Up to four graphs can be displayed at one time), and *Insta-Vestor* (calculates, defines, compares and graphs stock movement characteristics).

The *Management Combo* program integrates under a master menu with help screens the following three programs, also available separately: *Insta-Writer* (word processor), *Insta-Mail* (mailing and listing) and *Insta-File* (database management).



Database manager for ages 9 up. READER SERVICE NO. 69



Bubble Burst*—good clean fun. READER SERVICE NO. 70*

Insta programs are marketed by Microsci Marketing, a division of Standun Controls, Inc., 2158 Hathaway St., Santa Ana, CA 92705.

• Newsnet, the nation's largest specialized electronic newsletter publisher, has announced the addition of five additional investment newsletters and business publications.

The new business publications include *The Business Computer* (also known as "The Bible of Silicon Valley"), *Daily Metals Report, Fintex All-Day Foreign Exchange Monitor, Energies, Trends & Cycles, The Stanger Report* (tax shelter investments) and *The Wall Street Monitor.*

Newsnet began its operations in 1982. It operates at a 300 and 1200 baud rate and offers more than 150 business newsletters. The standard on-line usage rate during workday hours is $24 an hour.

Newsnet, 945 Haverford Rd., Bryn Mawr, PA 19010.

EDUCATION

For parents who see their computer as more than a toy, and for kids who must earn the right to play *Attack of the Mutant Camels*...we present a review of some—just *some*—some, mind you—of the latest education programs.

Secret Filer is the latest in Scholastic's Microzine series to be offered for the C-64. *Secret Filer* is a simple database management program for ages nine and up. The program will allow the user to enter and organize all types of information—lists of friends and enemies, facts, scraps from various interests, etc. $19.95.

Secret Filer joins *Adventure Double Feature, Mystery Double Feature, Logo Robot,* and *Poster* as spring releases from Scholastic.

Scholastic Inc., 730 Broadway, New York, NY 10003.

Spinnaker, arguably the leading developer of quality software for education, has produced *Bubble Burst,* in cartridge format for the 64. The game asks players four

to eight years of age to help Soapie the Sea Serpent finish her bath before the Zeboingers can spoil everything. The Zeboingers, little birds, fly in the window and burst Soapie's bubbles and try to turn on the shower. Using the joystick, the child tries to enclose the birds in bubbles so that they'll float away and leave Soapie in peace. Two skill levels cover the age spectrum. $39.95.

THE EVE OF THREE INTERFACES

The folks at Omnitronix have introduced three interfaces for both the 64 and the VIC.

• A Volksmodem Interface Cable. Anchor's Volksmodem is a low-cost modem, aimed at all major computers, which requires an interface cable. Since the 64 and VIC input/output signals are not directly compatible with standard modems anyway, complex circuitry would be required in addition to an interface. This interface cable comes complete with those modifications as well as a manual and a type-in BASIC terminal program. The cable, modem not included, sells for $22.95.

• An RS232 Interface Cable. A two-foot cable from the interface ends in a standard male DB25 connector, and the interface supports S232 pins 2 through 8, 20, and 22. A J-Cat type modem can be supported by adding an optional connector. Documentation includes a type-in machine language printer driver, BASIC dumb terminal program, and instructions on how to list a BASIC terminal program to the serial printer. Note that the interface is not intelligent; it will not convert the Commodore control characters to standard text in listing or printing. A money-back guarantee. $39.95.

• A Cassette Interface and Duplicator (CID). Plugging this interface into the cassette port, the user can use a standard portable cassette recorder to load and save VIC and 64 programs. Tapes will be fully compatible with a datasette and vice versa. High quality copies of games can be created through the CID by connecting the two recorders. An external switch on the CID can be tripped to allow the unit to be set for reverse polarity cassette recorders. $34.95.

Omnitronix, PO Box 12309, Seattle, WA 98111.

SECURITY BLANKET

Perhaps it's not a coincidence that we're living in an age of technological wonders *and* rampant crime; it may be that the lure of the former creates the latter. In any case, Jance Associates has created two home security systems for owners of the C-64 and VIC-20.

The Hard Wire system ($195) is a perimeter security system which uses magnetic switches on doors and windows.

The Wireless System ($349) is a line carrier system. Pulsed signals are sent over existing home electrical wires. The computer monitors motion detectors which send a signal if they sense a disturbance. If the system is not soon deactivated, the computer ignites the alarms.

Both systems can be exapanded to operate lights, appliances, furnaces, water heaters and so forth, according to a preset time schedule in the program.

Jance Associates, 346 Par Causeway, Wescosville, PA 18106.

FUZZBALL AND SO FORTH

"The Adventure Games That Took 3,000 Years to Create!" is Davka's boast. Davka Corporation creates games and tutorials primarily for the Jewish community, but many of their games will appeal to Gentiles as well. Their *Bible Baseball* is reviewed in this issue, but in case you're not famiiliar with Davka's other programs, here is a brief rundown. All are for the 64 only,



Three new interfaces from Omnitronix: RS232 Interface Cable, CID, and Volksmodem Interface Cable. READER SERVICE NO. 71

disk drive required.

The games include *IQ Baseball* (baseball trivia questions drive the nine-inning contest), *Jewish IQ Baseball* (questions on the Bible, history, Jewish festivals, Israel, etc.), *All about Chanukah* (for ages ten and up), *Fuzzball* (scoot across the floor and pick up the fuzzballs), and *Game of the Maccabees* (combat and a quiz).

MENU ANYONE?

Dynatech's latest entry in their CodeWriter library is *Menu-Writer;* it allows users to load and run programs from automatically generated screen displays.

MenuWriter can create up to 20 different screen menus, each showing up to 15 directory entries. Touching one key on the keyboard then loads and runs any program listed.

To set up the system, the user merely tells *MenuWriter* how the files are to be indexed—numbers, letters, cute codewords, whatever. No knowledge of file structures, programming, or operating systems is needed. *MenuWriter* translates the English instructions into the computer-language code. For the 64 only. $29.95.

Dynatech Microsoftware Inc., 7847 N. Caldwell Ave., Niles, IL 60648.

CHEATSHEETS

Don't turn your back on Leroy. Even the name *Cheatsheets* is misleading. Unless you are some kind of masochistic or macho user/programmer, *Leroy's Cheatsheets* from Cheatsheet Products are perfectly acceptable aids.

Leroy's Cheatsheets are plastic coated keyboard overlays for the 64 and VIC, compatible with a particular program only, that group together commands and



Leroy's Cheatsheets ***are stored on your computer, not up your sleeve. READER SERVICE NO. 72***

functions according to their use, display program starting functions and marked functions keys and additional tables, charts or drawings.

The Cheatsheets are available for the following VIC-20 programs: *Vicmon, Programmer's Aid, Super Expander, VIC Typewriter, VicTerm I, Quick Brown Fox, UMI Wordcraft 20, HES Writer, HES VIC Forth, Graphic Printer, BASIC.*

For the 64: *Term 64, Easy Script, HES Writer, Quick Brown Fox, Wordpro 3/plus, Graphic Printer, Paper Clip, Script 64, CalcResult, Hesmon, BASIC.* Each *Cheatsheet* is $3.95.

Cheatsheet Products, PO Box 8299, Pittsburgh, PA 15218.

COVER ME

If your VIC or 64 has a tendency to collect dust, jelly, coffee or needs a shield against frustrated pounding, you might want to consider the CompuCover, a high-impact plastic keyboard cover.

The easily-detachable clear cover not only acts as a dust and whatnot protector, but it can be flipped up to to prop manuals or input copy.

CompuCover, PO Box 2299, San Jose, CA 95150.

NO CLASS

Here is a short list of new software that fall between the classification cracks; that is, they didn't fit easily into any other news items but definitely merit a mention.

• D & M Software Publishers has released *Hi-Q,* a cassette-format program for the 64 that allows the user to create multiple choice quizzes on any subject the user chooses.

The user creates and stores the quiz, and D & M claims that "correct answers are rewarded with humorous antics, wrong answers yield humorous insults." $19.95.

D & M Software Publishers, 1510 South 97th St., Tacoma, WA 98444.

• *Lifestyle Budgeting* is Culverin Corporation's home budgeting program; the disk comes with a copy of Dr. Harper Roehm's *Spending Less and Enjoying It More* (published by McGraw-Hill) and a user's manual.

Culverin emphasizes that the program is a forecasting and modeling tool, not a checkbook balancer. The program takes a few hours to set up and then one

or two hours a month to monitor. The program assembles family financial records, distributes annual income into 12 increments, allocates annual expenses, performs revisions easily, performs calculations automatically, offers three planning options, and monitors each plan's savings projections table using color graphics and pie and bar charts. $49.95. 64 only.

Culverin Corporation, PO Box 503, Dayton, OH 45459.

• *Typing Tutor III,* from Simon & Schuster, was developed by Kriya Systems, Inc. for the 64. It employs Kriya's proprietary (hush hush) "time response monitoring" technique: each time the student engages the program, his or her responses are recorded. The tutorial's drill is then modified in subsequent lessons. The program uses the high-speed FORTH language rather than BASIC, which makes for a "quantum leap in responsiveness to the individual student's needs," according to Sat Tara Singh Khalsa, president of Kriya.

The program uses menus, explanatory prompts, and on-demand help documentation—such as an image of the keyboard onscreen—to wean the student off looking at the keyboard. Bar charts display speed and accuracy progress, and there *will* be a quiz. $49.95.

Simon & Schuster, 1230 Avenue of the Americas, New York, NY 10020.

• for both the VIC and the 64, *Blackjack Teacher* from Sota Enterprises simulates a blackjack table and teaches strategies that will allow the user to clean out the casinos.

The VIC version ($19.95) offers basic point counting advice; the 64 version ($59.95) offers basic to advanced lessons.

Sota Enterprises, 833 Garfield Ave., Suite 101, South Pasadena, CA 91030.

GAMES UPDATE

From T&F Software come the following new releases for the C-64:

Jet Star—you control the jet pack of Captain Twinkletoes as he gathers various minerals from the surface of a not-so-friendly planet.

Spogo—help a jittery critter jump from platform to platform while avoiding the bouncing balls that block his way.

Leafer Madness—become a caterpillar desperately beating feet towards the safety of his cocoon.

All of these games are in both cassette and disk format.

T&F, 16742 Stagg Street, Suite 106, Van Nuys, CA 91406.

From Broderbund comes the cartoon-animated, fun-filled *Spare Change.* In this antic adventure you are the owner of a very successful arcade. However, a pair of Zerks want to scarf up all of your tokens and use them for their retirement fund. It's your task to distract them long enough to collect all the tokens. When you fill any of the coin boxes, you'll be treated to an amusing cartoon featuring the Zerks themselves. On disk.

Broderbund, 17 Paul Drive, San Rafael, CA 94903.

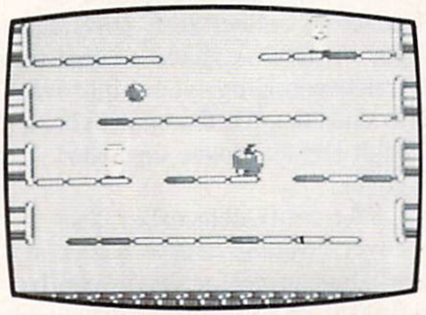
The first of Cadmean's "Discover America" series, *The Voyage of the Mayflower* uses graphics, 17th-century music, and realistic weather and sailing conditions to simulate the 1620 voyage of the Pilgrims to the new world. With each increasing difficulty level, more historical, geographical, and meteorological knowledge is needed. For the 64.

Cadmean Corporation, 309 Koch, Ann Arbor, MI 48103.

From Parker Brothers come 64 and VIC adaptations of their *Frogger* (cross a highway and river fraught with dangers), *Popeye* (save your sweet patootie from Bluto), *Q*Bert* (the kiwi-shaped hopping whatzit), *Star Wars* (fly down the throats of imperial fighters), *Gyruss* (pilot through deadly space), and *James Bond* (battle SPECTRE for queen and country).

Parker Brothers, 50 Dunham Road, Beverly, MD 01915.

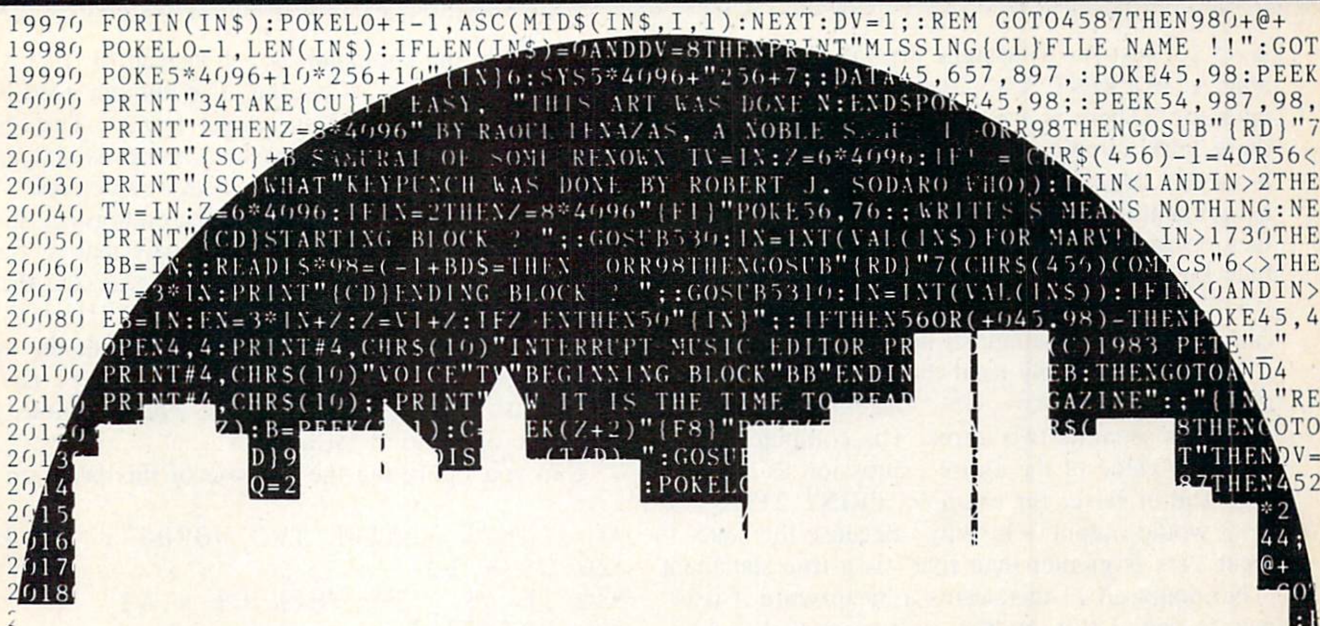






New games from T&F Software: Jet Star, Spogo, ***and*** **Leafer Madness,** ***all for the C-64 on cassette or disk.***
READER SERVICE NO. 73

RUPERT REPORT



THE CITY OF TRUTH

By Dale Rupert

You are standing at a fork in the road. You are allowed to ask only one question of the Guardian of the road in order to determine which path leads to the City of Truth. The only problem is that you don't know whether the Guardian is a Truthsayer (from the City of Truth) or a Fibber (from the City of Lies). The former always tells the truth and the latter never does. What will your one question be?

This is one of numerous similar problems in logical reasoning. It is interesting to speculate on whether or not there is, or will be, a computer that could arrive at the answer to this puzzle. As you try to figure it out, think how a computer might be programmed to do the same thing.

The computer is more than a word processor and a calculator. Its ability to perform logical operations and to modify its behavior as the result of logical conclusions is its primary distinguishing characteristic.

We are going to look at logical operators and logical expressions. If you have been programming for a while, you have been using these concepts, whether you realize it or not. We will explore the realm of computer programming in which the results are not really numbers and not really words. Logical expressions take one of two outcomes: true or false.

Without using your computer, predict the output from this statement:

```
A = 5
```

Or how about this one:

```
A = 0
```

You might be tempted to say that a Syntax Error would result; but, no, both statements are syntactically correct. Another guess might be that the computer will evaluate A and print its value, ignoring anything after the A. Sorry, wrong again. Okay, type it in and see what happens. The answers are 0 and −1, respectively.

Now where's the logic in that? Well, that's exactly what it is. The Commodore 64 (as well as many other computers) stores the logical values True as −1 and False as 0. The computer treated the "A=5" as a logical expression. Unless A had previously been given the value of 5, then the expression "A=5" is obviously false. Consequently the output of the PRINT statement was the value of the expression which is 0 (false). Similarly the expression "A=0" is true when the computer is first turned on. Thus the PRINT statement produced the value −1.

The equals sign in the above examples is not the same as the equals sign in a LET statement. LET

A=5 (or just A=5) assigns the value 5 to the variable A. In PRINT A=5, the "=" is a relational operator. (There is a lot of fancy nomenclature for these fairly simple concepts!) The term on the left is merely compared to the term on the right. No assignment or exchange of values is implied. In fact, some computer languages use different symbols for the assignment "=" and the relational "=" so the computer and the programmer alike can more easily distinguish between them.

You have probably used the other relational operators: >,<,>=,<=,<>. Once again, these operators separate two terms. The computer interprets the value of the entire expression as being either true or false. for example, PRINT 2*3 > 2+3 would output −1. Why? Because the statement "six is greater than five" is a true statement.

No doubt all of this seems a bit obscure if it is new to you. Let's see how well you are doing. Predict the output of each of these lines.

```
•10 PRINT 2 > 3
•20 PRINT 8 + 3 = 3 + 8
•30 PRINT A < A*2
•40 PRINT 5 + (B = 0)
•50 PRINT 5 + B = 0
```

Line 40 requires some careful thinking. The expression in parentheses is evaluated to be −1 (true). The rest is math: 5 + (−1) is 4. Do you see why line 50 would output 0? In that case, 5 + B would be evaluated to be 5 (since B is 0). The expression 5 = 0 is obviously false, so the printed value is 0.

Why would a statement such as line 40 ever be used? Suppose you are writing a game in which a bonus is to be added to the player's score after rounds 2 and 10. One way to do this very simply is to use logical expressions:

```
•10 SC = SC + (-20)*(RD=2) + (-30)
*(RD=10)
```

Here SC is the score and RD is the current round. If (RD=2) is true, then −20 is multiplied by −1 giving 29 which is added to the score. If (RD=10) is true, its value of −1 is multiplied by −30 to give 30 which is added to the score. In any other rounds, both logical terms are false, and a value of 0 is added to the score. Of course, there are other ways of giving the same results, but once you understand logical expressions, this is a very handy method.

Is there some reason that the logical value True is stored in the computer as −1 instead of +1? A rhetorical question, of course. The number zero is stored as sixteen zeroes in integer format. The opposite (or complement) of sixteen zeroes is sixteen ones, since the computer stores only zeroes and ones. The number stored as sixteen ones is the integer value −1. We won't go into why that is, but it is clear that the opposite of false is true and the opposite of (0000 0000 0000 0000) is (1111 1111 1111 1111). The computer treats these numbers as the integers 0 and −1 respectively. The integer +1 is stored as (0000 0000 0000 0001), which obviously isn't the opposite of false.

Can you figure out the purpose of this program:

```
•10 INPUT "ENTER TWO WORDS"; A$,B$
•20 C$ = B$
•30 IF A$ < B$ THEN C$ = A$
•40 PRINT C$
```

Predict what would be printed in line 40 if you entered CAT, DOG in response to line 10. This is a simple alphabetizer. Hopefully you are now able to recognize the logical expression in this program. It is the "A$ < B$ in line 30. The IF statement is always followed by a logical expression. If the first word you type comes before the second word alphabetically, then the expression A$ < B$ is true and the statement following THEN is executed. Otherwise C$ retains the value of B$, and line 40 prints whichever of the two words is first in the dictionary.

Line 20 is a handy way to eliminate a cumbersome GOTO statement in line 30. In line 20, we assume B$ will be printed. If A$ should be printed instead, we straighten it out in line 30. If we did not give C$ a value in line 30, our program would look like this:

```
•10 INPUT A$,B$
•20 IF A$ < B$ THEN C$ = A$ : GOTO
40
•30 C$ = B$
•40 PRINT C$
```

The program flow in the first example is smoother than it is in the second example, don't you agree?

Type this short program and run it:

```
•5 REM >> TRUTH TABLES <<
•10 REM   TRUE = -1 : FALSE = 0
•20 PRINT " A"," B", "(A OR B)"
•30 READ A,B
```

```
40 IF A=99 THEN END
50 PRINT A , B, (A OR B)
60 GOTO 30
70 DATA 0,0,0,-1,-1,0,-1,-1,99,99
```

The output is a truth table of the logical OR operator. Notice that all results are true except when both variables are false. If either A or B or both are true, then (A OR B) is true. Change the OR in lines 20 and 50 to AND. (A AND B) is true only when both variables are true. Now put NOT in front of (A AND B) in lines 20 and 50. This is a truth table for the NEGATED AND or NAND operator. Do you see how the NAND output is similar to the OR output? Changing the AND back to OR will produce the NOR truth table: A NOR B = NOT(A OR B). The output is true only when both inputs are false. Conversely the output of the AND operation is true only when both inputs are true.

Let's apply some of this abstractness to a simple, real-world example. Look at the following program. This program allows you to define the logic value of three statements. It then determines their logical conclusion.

```
5 REM RUPERT REPORT
10 REM LISTING 1 - LOGIC1
20 PRINT CHR$(147) :REM CLEAR SCR
EEN
30 TRUE=-1:FALSE=0
35 REM=============================
=======
40 REM >> IF  A$ , OR IF B$ AND C
$,
50 REM >> THEN T$
60 REM >> OTHERWISE F$
65 REM=============================
=======
70 A$="IT IS RAINING"
80 B$="IT IS SUNNY"
90 C$="THE TEMP. IS ABOVE 100"
100 T$="WE WILL STAY INSIDE"
110 F$="WE WILL GO OUTSIDE"
115 PRINT"IF "A$" OR IF "B$" AND
"C$", THEN "T$
120 PRINT:PRINT"--RESPOND T OR F-
-"
130 PRINTA$;:INPUT A1$
140 PRINTB$;:INPUT B1$
150 PRINTC$;:INPUT C1$
160 A=TRUE:B=TRUE:C=TRUE
170 IF A1$="F" THEN A=FALSE
180 IF B1$="F" THEN B=FALSE
190 IF C1$="F" THEN C=FALSE
200 D$=F$
210 IF A OR (B AND C) THEN D$=T$
220 PRINT D$
230 PRINT:PRINT"(PRESS ANY KEY TO
 CONTINUE)"
240 GET Z$:IF Z$=""THEN 240
250 GOTO 120
```

Lines 40 thru 60 show the format of the statement which is evaluated in line 210. The hypothesis is that "if it is raining, or if it is sunny and the temperature is above 100 degrees," and the conclusion is that "then we will stay inside." The terms are defined in lines 70 thru 90 as A$, B$, and C$. The conclusion is either T$ or F$ depending on the values of the terms.

Line 115 prints out the statement. Lines 130 thru 150 request the value (T or F) of each term. Just as in the previous example above, values are first assumed for A,B, and C in line 160. Their values may be changed in lines 170 thru 190, depending on your answers in lines 130 thru 150. Similarly, D$ is given the false conclusion value in line 200. If the conclusion should be true, it is redefined in line 210 where the whole expression is evaluated. Finally line 220 prints the conclusion. Line 230 gives a prompt. Line 240 waits for you to press a key to continue (notice the logical expression there). Line 250 jumps back to let you specify different weather conditions.

Lines 230 and 240 are desirable, since (on my Commodore 64, at least) the STOP key doesn't function if the program is waiting for a response to an INPUT statement. Try pressing the STOP key instead of T or F in line 130 and see if your computer behaves the same way. My computer ignores the STOP key—certainly an undesirable situation. Consequently the GET statement is used. The STOP key works with it.

It should be easy to modify this program for any logical hypotheses and conclusions. Simply redefine the terms in lines 70 through 110, and rewrite line 210. Adding another term would require including new lines similar to lines 130, 160, and 170.

If you thoroughly understand logical expressions now, or if you are willing to spend some time studying this article and are willing to work some examples, then you should be able to write even more sophisticated programs than before.

As an exercise, modify the program to corres-

Continued on page 77

THE COMMODORE PUZZLE BOOK: BASIC BRAINTEASERS By Gordon Lee and Nevin Scrimshaw

This $7.95 paperback is perfect for the person who delights in programming. The puzzles are mostly recycled from older collections of brainteasers, but the approach is fresh and exciting. So many computer books speak to a passive reader. This one demands that the reader wake up and actively engage his brain. The puzzles are challenging in themselves; the additional challenge of constructing a program to calculate the solutions will give the skills of even very experienced programmers a workout.

The book presents fifty problems. Simple black-and-white drawings help explain some of the puzzles. The second section of the book gives program listings for each puzzle. These BASIC programs are not the answer to the puzzle, or even the whole program that will solve the problem. Rather they are suggested alogorithms that point in the right direction, in case the reader gets stuck.

Beginning programmers may find this book frustrating. Anyone who is hazy about how to use functions such as STR$ and VAL will need to bone up some, but the authors assure us no advanced math is needed. The person who is uncomfortable with mathematical reasoning might find some of the puzzles a little alien at first. However, once the intermediate or advanced programmer figures out an approach that can yield a solution, the actual programs are short and fairly simple. The value of this book is the exercise it gives in approaching the computerization of problem solutions.

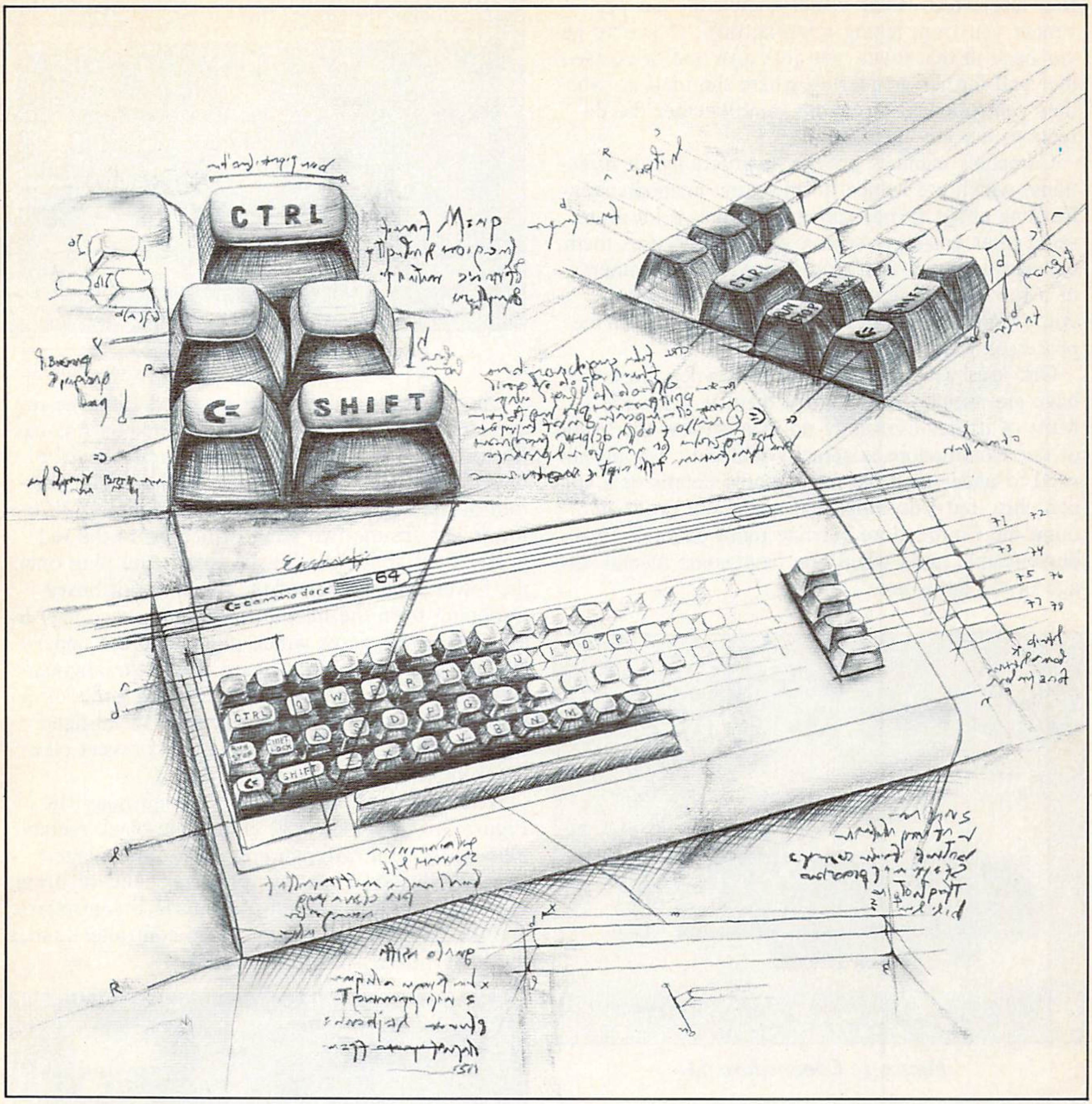
The puzzles include math problems, logic exercises, and cyrptograms. They are not grouped in order of difficulty. At first reading, none have obvious programming solutions (at least, no solutions that work). The careless reader will find that the easy, colloquial style of the book hides a carefully expressed problem. Since the tricky question is a time-honored mechanism in brainteasers, the reader can count it as all part of the fun.

The book title specifies Commodore, but these problems can be tackled on any home computer. The only statements peculiar to Commodore in the program listings seem to be PRINT formatting statements. The puzzles are engaging, and of just the right difficulty to while away an idle hour or so at the computer. The book can be dipped into, and savored in small doses. Kids will love it, and use the book to challenge each other in and out of school.

Anyone who works through all fifty problems will improve his problem-solving skills, sharpen his programming knowledge, and have a good time while he's at it. Teachers and students, experts, and fledglings just past novice class will all find something in it that meets their needs.

Birkhauser Boston, Inc., 380 Green Street, Cambridge, MA 02139.

—Annette Hinshaw



Anatomy of the Commodore 64

By Richard Herring

Recent microcomputer purchasers are typically befuddled by the new terrain they encounter in the world of computers. Although some of the landmarks, such as the TV and the keyboard, are familiar, many more are strangely unfamiliar or totally hidden from view. Most of the manuals we read are non-visual roadmaps, textual descriptions of how to get from here to there. If you have ever given or received instructions on how to get to a particular address in your town, you know how much visual imagery helps.

If someone is giving you directions in a town you know well, you can visualize the streets and buildings that are familiar. You may be told to turn left at the third light then drive past the mall, but when you get to the third light, you realize that you're at a two-lane street and the mall is on a

four-lane street. Your visual knowledge has prevented you from taking a wrong turn. If you've never been in that town, you just turn and drive farther and farther wondering where the mall is, whether you followed directions, and whether the directions are any good at all.

Computer manuals provide instructions, or directions, which are subject to the same phenomenon. Reading about sprites somehow makes a lot more sense after you have seen a game which uses them, or even designed and programmed the movements of a few. For programming graphics and sound, you eventually will develop mental images for the processes which occur.

One area where most computer owners do not have any visual data to guide them is hardware. Most of us have virtually no idea of how the inside of our Commodore is actually laid out. You do not need to understand the microscopic details of a silicon chip, but I do think that using and programming the Commodore become more concrete, understandable tasks when you have some mental image of the hardware.



Figure 1: Commodore 64

Now, you could run out to the garage, grab a few screwdrivers (or a hacksaw if you attended the Genghis Khan School of Delicate Electronics) and rip your Commodore apart, then get a schematic and match the circuit board locations to chip numbers, and dig through technical manuals to figure out what each chip does. Of course, you would kill your warranty along the way. Or, you can keep reading and come with me on a vicarious tour of a Commodore 64. We'll picture and describe a purely hypothetical machine so nobody's warranty will bite the dust. Remember too that in-house design changes occur (usually reducing the number of components), so every Commodore 64 will not be identical inside, but the major functional components we'll look at and describe are pretty universal.

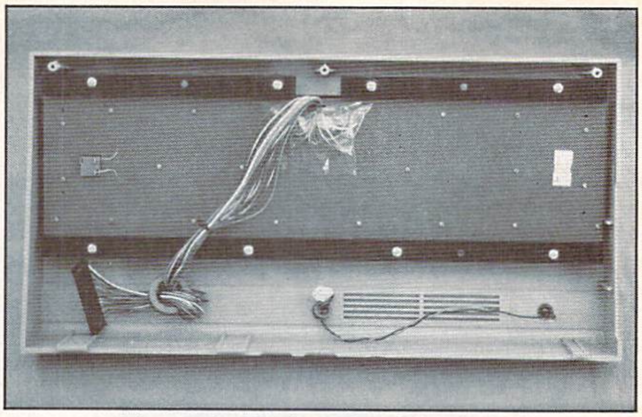


Figure 2: Underside of keyboard

Once the three screws are removed from the underside of the Commodore 64, the keyboard is easily lifted off. It is attached to the circuit board of the computer only by two sets of wires. The bottom of the keyboard is pictured in Figure 2. In the lower left are the two wires which go to the red power light above the function keys and plug onto the lower right corner of the main circuit board. Dropping from the middle (under the space bar) is a bundle of 18 wires which plugs onto the upper left corner of the circuit board. Once the keyboard and case top are removed, the Commodore's circuit board remains covered by a piece of metal-lined cardboard which acts as shielding to prevent electrical interference.

When you look at the C-64's circuit board in Figure 3, you will see 32 chips—the black rectangles—in several different sizes. The four biggest ones are the 6510 central processing unit, or brain; two 6526 CIAs, or peripheral interface controllers; and the 6567 VIC, or color video controller (under the lower metal shield; see Figure 4).

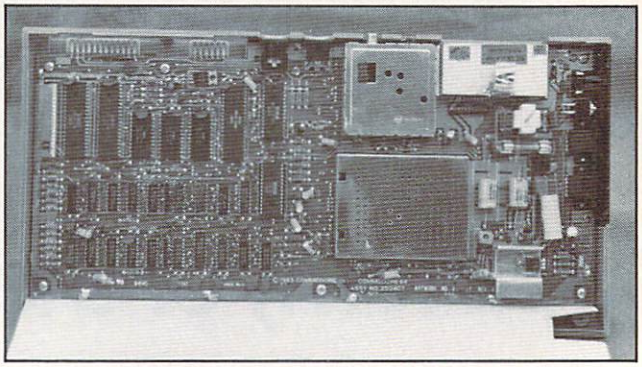


Figure 3: C-64 circuit board

The 6510 chip is the sixth chip over from the upper left corner of Figure 3. As the brain of the computer, it is actually responsible for changing and moving the information stored in memory. It handles eight bits (or BInary digiTS) at a time, which is why the Commodore is considered an 8-bit computer. Since each bit can have one of two conditions (represented by a "0" or a "1"), the

6510 can recognize individual numbers up to 255. In binary arithmetic 00000000 and 11111111 represent the base ten numbers zero and 255 respectively. How then, you ask, can my computer work with a large numbers? And especially, how can it address 64K bytes of RAM?

Let's start an explanation by remembering that we are working with base 2 numbers (0's and 1's). Although many people think of 1K as 1000 bytes, there is no power to which 2 can be raised that equals 1000. The closest base 2 number is 1024, or 2 to the tenth power. That is, in fact, exactly what 1K represents—1024 bytes. So your 64K computer actually has 65,536 bytes of RAM (64 times 1024). The number 65,536 is important when we consider that the 6510 brain of the Commodore can only work with one byte of data at a time, or recognize 265 discrete numbers (0 through 255). Yet, it can find its way around in those 65,536 bytes of RAM. It does that by using two bytes, back to back, to deal with numbers up to 65,536. Each byte represents 256 possible numbers and 256 times 256 equals—you guessed it—65,536.

It's a lot like your finding a new street address—say, 311 Elm. Imagine that you can only remember three characters at a time. So your directions tell you to find Elm (the first byte for the computer). That gets you in the general vicinity. Then you can forget Elm when you get the next instruction, 311 (the second byte for the computer). You just move along Elm until you get to 311, and you're there. In just the same way, the computer's 64K memory can be visualized as 256 streets with 256 houses on each street. Computer manuals will actually call each street, or 256 bytes of memory, a page of memory.

Without going into a lot of detail, the 6510 central processing unit (CPU, or MPU if you prefer the term "microprocessing unit") is a simple chip as CPUs go. It understands 56 different instructions in assembly language, the computer language closest to the machine's actual language. These are the 56 discrete tasks it can actually do, and bear no resemblance to the things you can do in BASIC. BASIC is really a program which interprets each command you type into a complex set of instructions for the 6510 CPU. The kinds of things the 6510 can do as a result of one of the 56 instructions it recognizes are tasks like fetching a number from a RAM address into the CPU, or adding a number from a particular RAM location to the number already in the CPU. Simple stuff, so how can the Commodore run all those neat, complicated programs? Speed is the answer. The 6510 can operate at 1 or 2 MHz (megahertz)—1 or 2 million cycles per second. Even when each instruction completes only a tiny step, you can move a lot of them through at that speed.

By comparison, the 6502 CPU made by MOS Technology and used in Apple and Atari computers has a very similar assembly language instruction set to that of the 6510, but operates between 1.2 and 1.8 MHz. The eight-bit chips in CP/M computers, however, have expanded instruction sets. Zilog's 8080 CPU has 78 instructions and its Z80 chip, used in the CP/M module for the Commodore 64, has 158 instructions. Certainly, the 6510 can do nearly anything the Z80 can do, but it may take many more instructions to get the 6510 to complete its task.

In the upper left corner of Figure 3 are two more of the C-64's four large chips. Both are 6526 CIA chips, called peripheral interface devices because they allow the computer to work with various peripherals like the keyboard and disk drive. The 6526's are the clocks of the system. As fast as information is passed around, accurate timing is a must. Imagine that the rush hour traffic in New York City all had to pass through one intersection in a fraction of a minute. Assuming that braking and acceleration were instantaneous, it might be possible—timing would be the key. In addition to coordinating the flow of data traffic in your computer system, the 6526's also provide a clock for real time applications. They count in four units of time—tenths of seconds, seconds, minutes, and hours—to provide a 24-hour clock. No, there is no digital display inside your computer. It's all those invisible 0's and 1's in the 6526. Software, however, can look into a 6526, read the 0's and 1's and display the changing time on your screen.

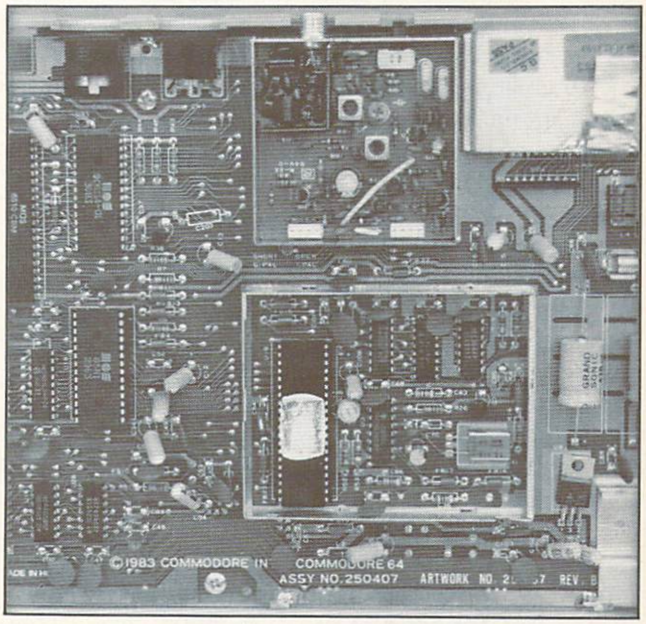


Figure 4: Closeup of shielded areas

The Commodore 64's fourth large chip is the 6567 VIC, or video interface chip. Although it is hidden by the lower shield in Figure 3, it is visible in the close-up in Figure 4. The VIC is described as a multi-purpose video controller used in compu-

ter terminals and videogames. This chip keeps track of screen positions by row and column when the screen scrolls and calculates light pen coordinates on the screen. One of the VIC's most important, though most complex, functions is handling interrupts. Interrupts can occur at four different times—when two sprites collide, when a sprite collides with something drawn on the screen, when a light pen is used, and between the 60 times each second when your TV picture is redisplayed. Much of the VIC's power, as far as we are concerned, is its ability to control sprites, graphics characters which can move independently of the screen display.

The VIC chip can use as much as 16K of RAM for information on screen display. Text—letters, numbers and special characters—are displayed on your screen in 25 rows of 40 characters each. Each character can be any of 16 colors with four background colors available at any one time. Characters are formed from a matrix that is eight dots high and eight dots wide. (Each dot is one pixel, or PIcture ELement, on your screen.) If you are willing to give up some resolution, the VIC can display each character as a four-by-eight-dot matrix (each dot is twice as wide, so characters remain the same size). One character can then contain two colors and have two background colors.

In addition to eight-by-eight blocks of pixels, the VIC can locate individual pixels anywhere on the screen. This gives you 320 dots across the screen and 200 down, or 64,000 individual pixels which can be controlled. Called "bit-mapping," the result is the high resolution graphics which are so dramatic to create on the Commodore. For each eight-by-eight group of pixels in this bit-map mode, the number of colors is limited the same way that character colors are.

Sprites are sometimes called movable object blocks, or MOBS. The Commodore 64's VIC can control eight sprites, each up to 24 pixels wide and 21 high. In addition to its own color, each sprite is transparent wherever its pixels, in that 24-by-21 pattern, are not used, so the background over which the sprite is moving can show through. Sprites can be expanded so they are twice as wide, but with no higher resolution. (You get 12 pairs of dots vertically.) Or, if you are willing to cut their width in half to 12 pixels, they can contain three colors. Although the screen is 320 pixels wide and 200 high, sprites can be moved through 512 horizontal and 256 vertical positions. With such a large range, sprites can move smoothly onto and off the screen.

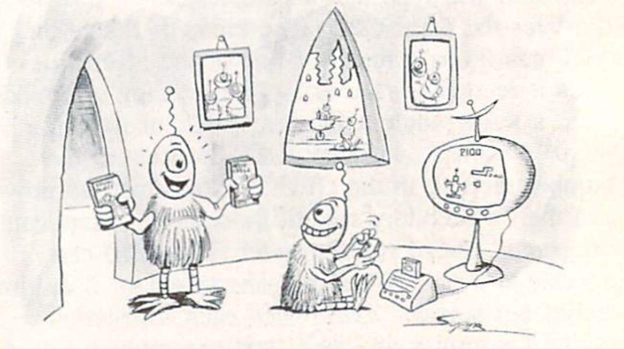
Sprites can not only pass in front of each other, but can also move behind other objects drawn on the screen. To determine whether a sprite will move over or behind part of a graphics display, the programmer sets the sprites priority. The priority of sprites to one another never changes; they are numbered from zero to seven. The fancy displays you can create with sprites do have their cost—1000 bytes of RAM.

The sounds that issue from your Commodore 64 are controlled by the 6582 SID chip. The SID, or sound interface device, is located near the middle of the circuit board, under and to the right of the 6510 CPU (see Figure 3). Capable of producing three independent voices, or notes, at a time, the SID acts as the Commodore's sound effects generator and music synthesizer. You are given a great of control over each sound produced. By programming the SID, you control each note's frequency (pitches from 0 to 4 kHz), volume, and tone color. Also called simple harmonic content, the tone color of each note can be set to any of four pre-defined waveforms.

For each of the three voices which the SID can produce, you can design what Commodore calls an "envelope." In other words, you can control how fast a note becomes louder (called attack), how fast a note becomes softer (called decay or release), and how long the note will be held at a particular volume (called sustain). Additionally, the SID chip serves two functions not related to sound generation. If you are willing to give up one of the Commodore's three voices, it can generate random numbers. Also, it can get position information from paddle controllers.

Random access memory, or RAM, is the Commodore's workspace. Although computations occur in the 6510 CPU, all the information which the 6510 uses and the results of all its computations are stored in RAM. As we saw before, the Commodore 64 actually has 65,536 bytes of RAM. These are contained in eight chips of eight kilobytes each, shown in the lower left corner of Figure 3. The 8K RAM chips are the second to fifth chips in the two bottom rows.

Undoubtedly, you have noticed that your computer does not seem to have 64K—at least not 64K



"We can play a normal game like 'Cosmic Blast' or a weird game like 'Baseball...'"

that you can use for your programs. That is because a number of other programs must reside in the Commodore's 64K of RAM. Somewhere, there must be instructions to the CPU on how to display characters on the screen, to accept input from the keyboard, and to send data to the disk drive. Also, unless you want to communicate with your computer in its own language of 0's and 1's, a programming language will sit in RAM.

Your computer's RAM is broken into nine major blocks. The Commodore's operating system sits in the first kilobyte of RAM and the second kilobyte is used to remember the contents of the TV screen. Normally available to you when the computer is turned on, the next 38K (3 to 40) are free space for your programs and data. Next, the BASIC language sits in 8K (41 to 48) and 4K are used for special programs (49-52). The video interface chip, VIC, and sound interface device, SID, use up 2K (53 to 54), and the color memory for each character position on the screen, 1K more (55). Input and output, handled by the 6526's, including the keyboard, RS-232 port, and CP/M module, take 1K (56). Finally, the Commodore 64's KERNAL operating system sits in the last 8K (57 to 64).

That is the standard configuration of RAM when you turn on your computer. You can, however, free up any of those blocks of RAM if you are willing to give up the functions which reside in them. By giving up access to BASIC, you gain 8K more RAM for your own programs and data. Of course, you would not want to give up some functions like the operating system. Although you could have 64K of totally empty RAM just waiting to be programmed, you would have no way to get data in or out.

What happens to these functions of the computer if you free up the RAM where they normally reside? For that matter, if information in RAM disappears when the power is turned off, where do these programs come from? They are all permanently placed in other chips in your computer.

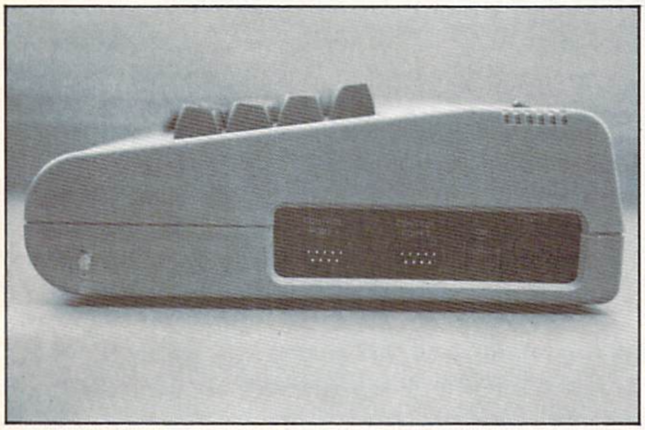


Figure 5: Side of C-64

These read-only memory, or ROM, chips retain their information even when the power is off. When you hit the power switch, the 6510 CPU automatically looks at the ROM chips and at some of the RAM. Why not look at all 64K of RAM? Remember that the CPU can only find its way around in (address) 64K of memory. That 64K limit applies to all memory—ROM and RAM. By putting BASIC and the operating programs on ROM chips and still providing 64K RAM, Commodore gave us the best of both worlds. We have an operating system and a programming language when we want them, but we can also turn them off to get more free memory.

There are several ROM chips in the Commodore 64. Three major ones, located between the 6526 CIAs and the 6510 CPU, are the third through fifth, medium-sized chips in the top row of Figure 3. The first of these chips contains the BASIC programming language; the second, the KERNAL operating system; and the third, the C-64's character set. Another ROM, the color ROM, is a small chip just left of the SID chip in the center of Figure 3.

In addition to the plugs for the power supply and TV cable, the Commodore 64 has seven ports through which it can be connected to various peripherals. On the right side (Figure 5), next to the power switch, are two joystick ports. Each of these can also be used with a pair of paddles, and the first one can be connected to a light pen. The rear of your computer (Figure 6) shows the remaining five ports.

Beginning at the left, we see the 44-pin expansion port where you can plug in various program modules. Through this port, you gain access to the computer's address and data buses, or data paths. That is why it is important for your computer to be turned off whenever you plug in or unplug a module. A charge of static electricity, built up in you or in the module, could travel directly through the

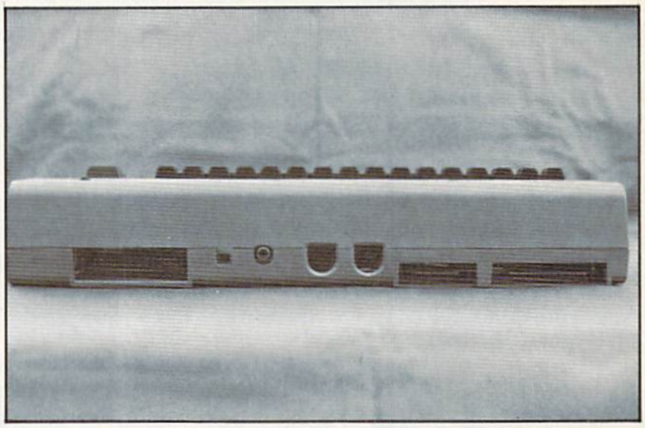
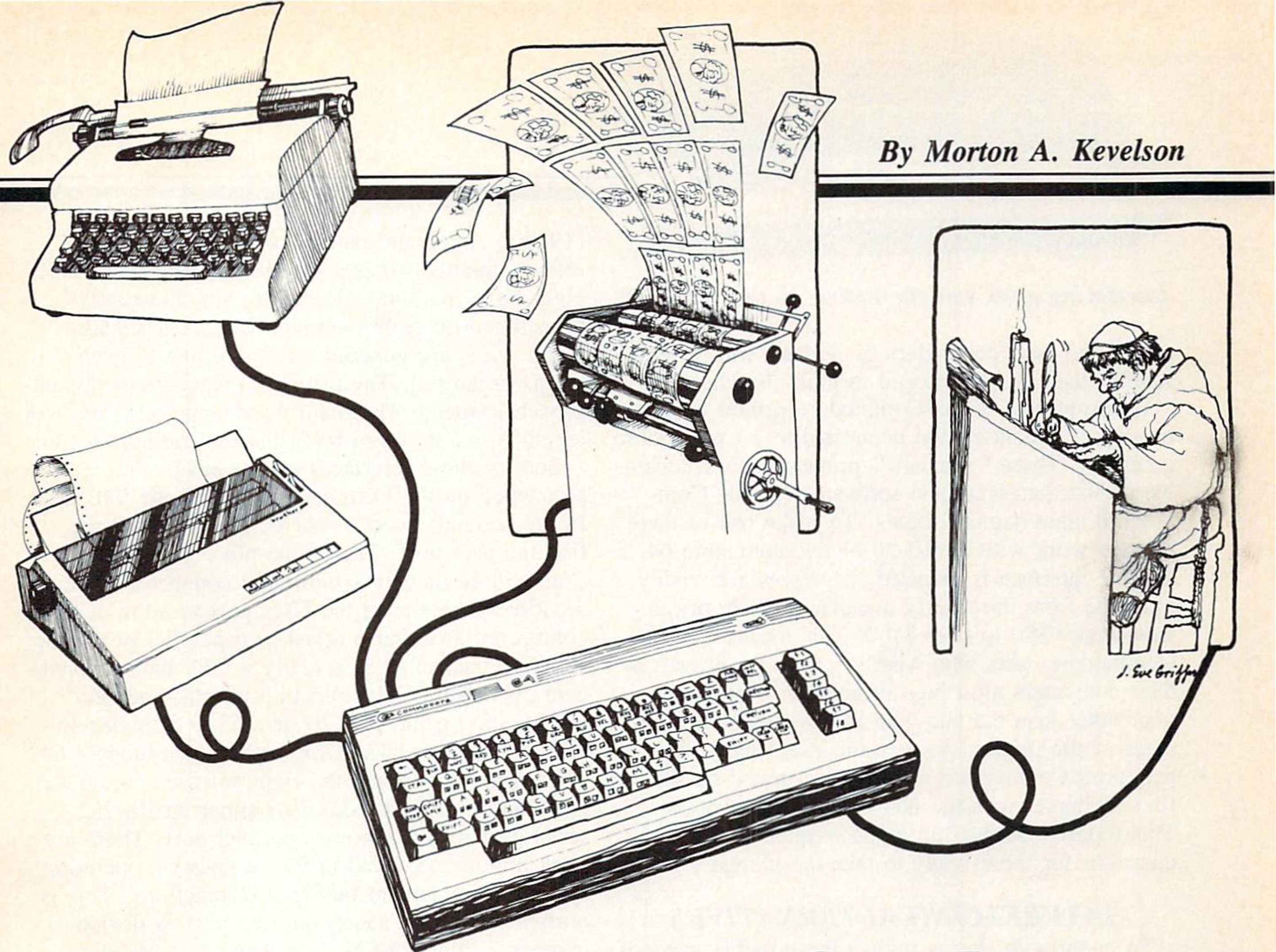


Figure 6: Rear of C-64

Continued on page 76

By Morton A. Kevelson

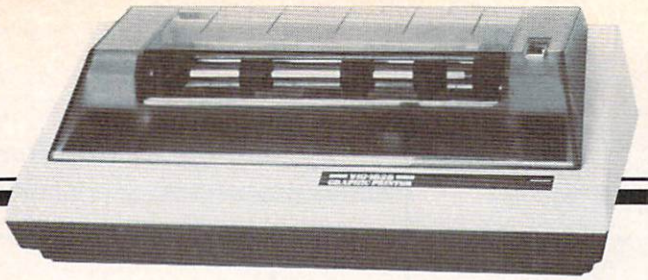


Printer Interfacing
for the Commodore 64 & the VIC-20

The importance of a printer for maximizing the utility of the home computer cannot be minimized. The availability of hard copy on demand is a tremendous convenience for some applications and a necessity for many others. One need only try debugging a 100 line by 8 column BASIC program on a 25 line by 40 column display to discover the value of a printed program listing. Any type of program development work benefits greatly from the availability of hard copy.

Applications such as word processing, mailing lists and data bases are virtually worthless without a printer. It is no wonder that after mass storage (tape or disk), the printer tops the list of home computer accessories.

In recognition of this fundamental need, Commodore offers a line of low-cost printers. For a price not much greater than the purchase price of a Commodore 64, the VIC-1525 and its successor the MPS-801 are available. These offer a reasonable compromise between price and performance. Even more important is the complete compatibility of the Commodore printers with the hardware and software requirements of the computers. Although satisfactory for casual applications, these Commodore printers lack many of the features and capabilities of the products available in the non-Commodore world. The first feature to be missed is the much higher printing speeds available on other printers. The 30 and 50 character per second speeds of the VIC-1525 and the MPS-801 are well behind the 120 and 160 characters per second provided by other printers. Add to this bidirectional printing capability and one can expect throughputs 6 to 10 times as great. If your applications require other features, such as superscripts and subscripts, alternate typefaces, or just plain better quality print



Interfacing gives you alternatives to the VIC-1525.

with lower case descenders or perhaps true letter quality, then you are forced to look elsewhere.

Unfortunately, a non-Commodore printer cannot be simply purchased and connected to a Commodore computer. These "standard" printers are incompatible in both hardware and software with the Commodore input/output formats. To make one of these printers work with a VIC-20 or a Commodore 64, a suitable interface is required. Interfaces are readily available from third party manufacturers in prices from under $20 to over $100. This means that Commodore users who wish to connect a printer to their computers must buy an interface from a supplier other than the one who makes either the computer or the printer. As a result, even many experienced Commodore users are frustrated by their first encounter with the non-Commodore world. This report will clear up many frequently asked questions for those ready to take the plunge.

INTERFACING ALTERNATIVES

To begin with, let us review the possible ways of connecting a printer to a Commodore 64 or a VIC-20 computer. From Commodore's viewpoint there is only one proper way, the IEEE serial port. That is the round, six-pin connector next to the cassette port, which is also used by the disk drive. For the rest of the world, two "standards" prevail, the RS-232 serial interface and the Centronics parallel interface.

The RS-232 standard is actually implemented in various ways. The common thread is the serial nature of the connection whereby data is transmitted in a serial stream one bit after the other. Along with the single data line there may be several additional wires to maintain the flow of information in an orderly fashion. The advantage of this format is relatively low cost in that a minimum number of wires are required. On the other hand, this interface tends to be an extra cost option on many printers. The signal format allows the physical length of the connecting cable to be rather long, on the order of some tens of feet. The primary disadvantage is slower transmission speed. The maximum transmission rate at the Commodore User port is 2400 Baud or approximately 2400 characters per minute.

The Centronics interface is really more a bit of history than a standard. It is named after an early (1970's) American manufacturer of printers for microcomputers. It is a parallel interface; that is, eight bits (one byte) of data are simultaneously transmitted on eight parallel wires. A few additional wires are generally included just to keep things organized. The primary advantage of this approach is speed. The main disadvantage is the short length (three to seven feet) that the cables can run.

Both of these interfaces can be easijly implemented on the Commodore 64 and the VIC-20. In the rear left hand corner of the computer we find the user port. This is the place where the Commodore modem is normally connected. Commodore has designed the User port to allow it to be configured as either a serial or a parallel port under software control. It is actually a fully bidirectional data port on a 6522 "versatile interface adapter" (VIA) chip on the VIC-20 or a 6526 Complex Interface Adapter (CIA) Chip on the Commodore 64. This means that given the right software, the User port can be made to look like either an RS-232 serial port or a Centronics parallel port. There are even routines included in the computer's operating system to implement the RS-232 functions. This is normally done by addressing the port as device number 2 in an OPEN statement.

Although the software to configure the port is relatively straightforward, the hardware takes a little more work. To begin with, the physical connector used by Commodore is a 24-pin printed circuit edge card connector. The standard RS-232 or Centronics connectors are the D type. This means that an adapter cable with matching connectors at each end will be required to complete the connection. These cables are readily available for under $20. For a little more you can purchase a software driver with the cable. The smart ASCII interface from Midwest Micro Associates is this type of adapter package for Centronics printers.

The RS-232 connection will require a little more work. Although the Commodore RS-232 is in the standard format, the output voltage levels are not. The output voltages at the user port are standard transistor transistor logic (TTL) which are from 0 to +5 volts. The RS-232 standard calls for a voltage range from −12 to +12 volts. Although many of the RS-232 interfaces supplied with the printers will function properly under these conditions, many others will not. Thus the cable which interfaces to an RS-232 printer should include the necessary electronics to account for the voltage difference as well as the physical connection. These adapters are not very expensive, generally on the

order of $30.

The main advantage of using the user port for printer interfacing is the low cost of the interface. The disadvantages, though, seem to negate this approach. To begin with, you cannot use the port for both a modem and a printer at the same time. Both the Commodore modem and commercial RS-232 versions must use the user port to interface with the computer. This precludes the possibility of downloading text to your printer while communicating with another computer. Software compatibility is somewhat more subtle but equally vexing. Although many commercial programs, such as *Easy Script,* allow for the possibility of non-Commodore printer connections to the User port, many other packages assume the use of a VIC-1525. Thus it is entirely possible to find yourself in the possession of a desirable software package which will not work with your printer connection on the user port.

In terms of writing your own programs, the software conversion is even more complex. To begin with, the proper routines must be included to properly configure the user port. This is a relatively minor problem in comparison to the conversion of the Commodore character codes to the ASCII codes recognized by most printers. The term ASCII is an acronym for American National Standard Code for Information Interchange. It describes a format using a coded character set consisting of seven bit coded characters (eight bits if parity check is included). The code is used for information interchange among data processing systems, data communications systems, and associated equipment (read the last one: printers). The ASCII set consists of control characters, the alphanumerics (letters and numbers) and punctuation.

As you might expect, the Commodore character code is not the same as the ASCII code. To begin with, the Commodore code uses a full eight bits. Those who know some binary math will immediately realize that this allows up to 265 unique characters to be defined. The seven bit ASCII code is limited to 128 characters. On top of all this, Commodore has two distinct character sets, with considerable duplication between them. The first set includes the upper case alphabet and all of the Commodore graphics symbols. The second set consists of the upper and lower case alphabet and the graphics symbols that are generated when the Commodore key is held down on the keyboard. Finally, all these characters can be printed in normal and reverse mode. To further complicate matters, many of the Commodore print control codes, such as the cursor keys, clear screen, etc., use the same codes as ASCII controls but mean different things. Thus, trying to send Commodore "ASCII" to a standard printer can yield very strange results.

The Commodore character codes are listed in Appendix J of the VIC-20 users manual, Appendix F in the *Commodore 64 User's Guide,* Appendix F of the *VIC-20 Programmer's Reference Guide* and Appendix C of the *Commodore 64 Programmer's Reference Guide.* Page 274 of the *Personal Computing on the VIC-20* includes the standard ASCII codes as well. Examining these tables shows that in the upper case graphics mode (Commodore's cursor up mode), the Commodore codes for the upper case characters correspond to the ASCII upper case codes. However, in the lower case upper case mode (Commodore's cursor down mode), the correspondence to the ASCII code has been reversed.

As if all this were not enough, how does one go about printing any of the Commodore's graphic symbols on a non-Commodore printer? If you are using a formed character letter quality printer, you don't. The character set of a letter quality printer is fixed by the daisy wheel or whatever mechanism it uses. If you have a dot matrix printer with graphics capabilities, you can. The procedure uses the high resolution graphics capability which most dot matrix printers are capable of. Each of the Commodore characters have to be custom programmed for printing when using the dot matrix graphics mode.

The result is that the conversion software must perform a large number of repetitive tasks, just what computers are for. In order to achieve any meaningful speed, machine language is essential. Even with careful coding, the resulting program will be rather long. Depending on the specific conversions you wish to implement, the resulting program will occupy several hundred bytes to one or two kilobytes. This is not a problem on the Commodore 64; however, the unexpanded VIC-20 can ill afford the problem of interfacing the printer with many commercial software packages, which may very well require the memory set aside for your printer routine.

Fortunately, there is another way. Have the interface cable itself perform all the necessary code conversions as well as the hardware adaptation. This is just the approach adopted by the Tymac Controls Corporation with their Connection Intelligent Parallel Interface and by Cardco, Inc. with their Card?/+G Printer Interface. For those of you

who are puzzled by the "?" in the Cardco interface names, just look up the shorthand entry form for Commodore's PRINT command.

Before we take a close look at these interfaces, a few words on how Commodore BASIC communicates with the peripherals is in order. The first step is to open a channel to the device by using the OPEN command. The format is:

```
OPEN LFN,DN,SA
```

where lfn stands for logical file number, which can be any integer between 0 and 255. Note that an lfn of 128 or greater will generate a line feed as well as a carriage return after a print statement. This is rarely needed, so keep your lfn less than 128. The second value, dn, is the device number, which can be 4, 5, 6, or 7 for the printer. This identifies the particular hardware device the computer will be communicating with. The last value is the secondary address, sometimes referred to as the command. The significance of this value varies with the device being addressed. In general, it will represent an instruction or a command to perform a particular function. Next month, we'll include a table of all the secondary addresses and their functions as they relate to the Commodore printers and the interfaces we will be looking at in this review.

Once a channel has been opened, data and further instructions are sent to a device via the PRINT# statement. The format is:

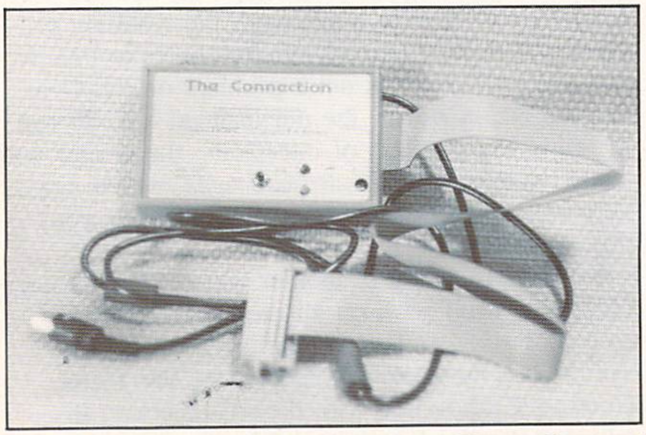
```
PRINT#LFN,DATA
```

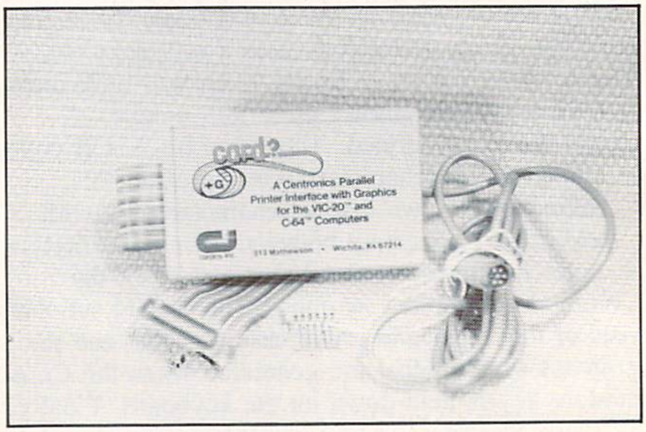
where lfn is the same value as was used in the open statement. The term "data" can be any valid Commodore data form, either a numeric or string variable or constant. Actually, PRINT# command can be thought to transmit data to a device the same way the PRINT command transmits data to screen. Note that the # sign follows the PRINT without any intervening space. There is one more BASIC command which we must review: the CHR$ function. The format is:

```
CHR$(NUM)
```

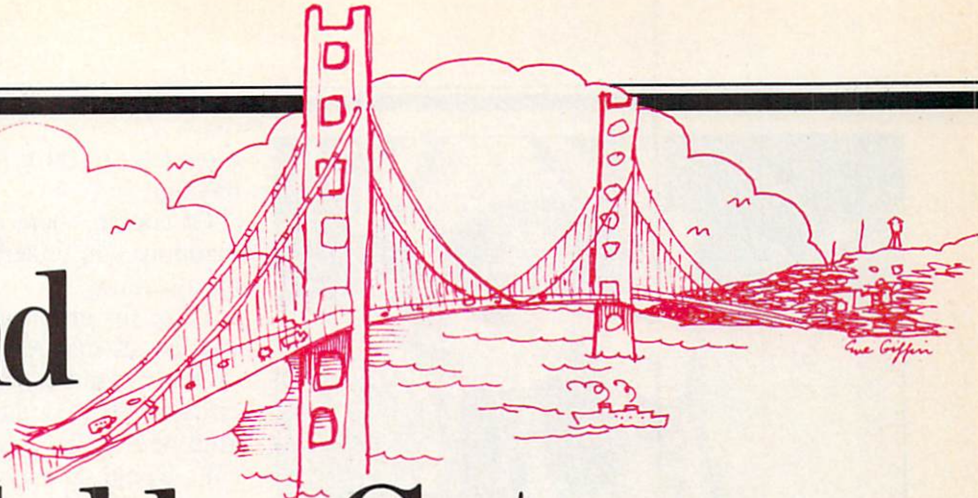
where num is any number or numeric expression which when evaluated will round down to a value in the range of 0 to 255. The resultant number will be converted to a string only one character long. This function is important as it is the only way to transmit certain characters in a print statement. For example, PRINT CHR$(8) will transmit to the screen a string one character long whose value is 8. This will have the effect of disabling the SHIFT/COMMODORE key. Note that a PRINT 8 or PRINT "8" will not have the same effect. These will merely print the character eight to the screen, not transmit a value of 8 to the screen. Also, PRINT CHR$(56) is actually the same as PRINT "8" since the character 8 has a Commodore character code of 56. It is important that you become familiar with these concepts as they are the key to effectively controlling your printer or, for that matter, any peripheral device.

This wraps up the background material on printer interfacing. Next month we will take a close look at two printer interfaces with graphic capabilities for the Commodore 64 and the VIC-20, the Tymac Connection and the new Cardco Card?/+G. we will present the results of an extensive series of tests using these interfaces with a Commodore 64 and a Star Micronics Gemini 10X high speed dot matrix printer. You'll find those results rather interesting!





Two Commodore interfacing options: Tymac Connection and Cardco Card?/+G, to be scrutinized next month.



Beyond the Golden Gateway

A reply to David Ritchie's **Golden Gateway** *in the February* **AHOY!**

By Stephen Bent

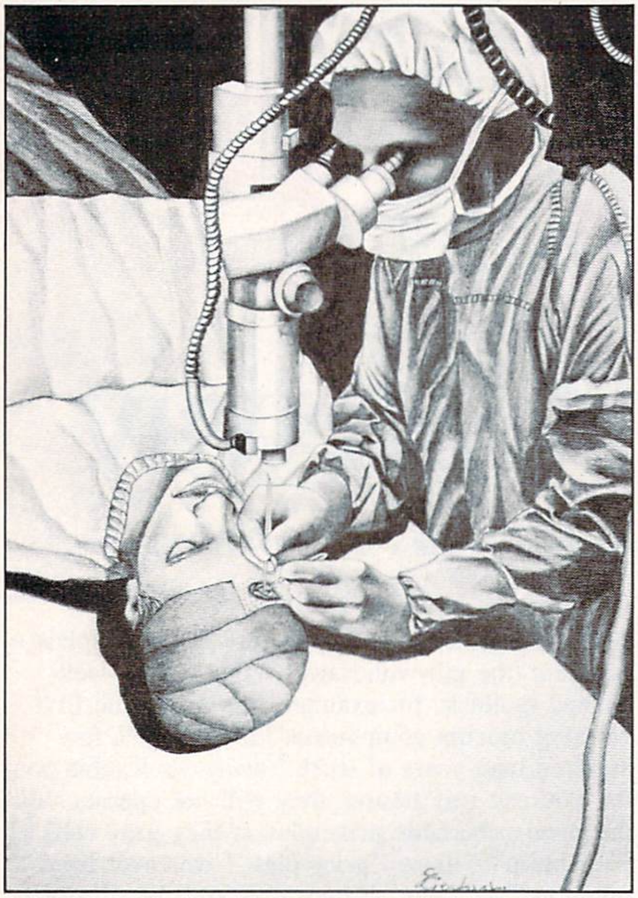
Michael Faraday, a 19th-century pioneer in the study of electromagnetism, once observed that "Nothing is too wonderful to be true." Twentieth century types, having witnessed the advent of the Nuclear Age, the Space Age, and the Biotechnology Age all within the span of a single lifetime, would hardly disagree.

So who can fault Mr. Ritchie for playing the modern day Coronado, exhorting us on to a biochip-accessible "Golden Gateway" that will swing ajar in the not-too-distant future to reveal golden technological vistas populated by molecular computers, "transmogs," and "telepresences"?

But before you start planning for a mind-machine linkup with your Atari 5200, keep two points in mind: first, Coronado died a disappointed man, never having found his City of Gold; second, molecular electronics, of which the so-called biochip is one manifestation, has yet to produce a single practical device, let alone convince a sizable number of skeptics in the field that organic versions of silicon- and gallium-based integrated circuits present a commercially viable alternative to existing technology.

Ritchie's rosy predictions aside, what are the prospects for a biochip in your future? And while we're at it—just what *is* a "biochip," anyway? Ritchie's definition—"a microprocessor built along biological lines"—and his repeated suggestion that a practical biochip would be a miniaturized variation ("improved," no less!) on some neuron cell network in a living organism is misleading, to say the least. First of all, we have only the most rudimentary notions of how neurons are organized into functional systems, and then only in the simplest of contexts (the gill-withdrawal reflex of one well-studied mollusk, for example). Even if "the first working biochip components" are only "a few hundred man-years of work" away, as Ritchie posits, you can rest assured they will *not* operate "on the same cybernetic principles as they gray cells in your brain"—those "principles," whatever they might be, are quite obscure now and, in all probability, will remain so for the forseeable future. (We shall return to this fact of life in a moment.)

So what are we talking about here? Generally speaking, the goal of researchers in the fledgling field of molecular electronics is to replace tiny solid-state devices (transistors and such), wherein the migration of charge carriers through crystalline lattices is regulated, with even tinier systems of interconnected molecules, each capable of assuming at least two distinct states ("on" and "off") that differ in some measurable conductive, magnetic, or optical property. The idea of building logic circuits out of such molecular "flip-flops" is hardly new—away back in 1960, a fellow named William Biernet obtained a patent for a memory unit in which portions of organic (that is, carbon-based) molecules underwent a spatial realignment (say, a rotation) upon the application of a combined electrical and magnetic field; the rotated configuration represented a particular logic state, for example "0" or "1." Some 20 years later, two scientists at IBM, Arieh Aviram and Philip Seider, developed and patented an "organic memory device" in which a switch in logical state (from "0" to "1" or "on" to "off") occurred with the movement of



an electron from one end of an organic molecule to another, thereby altering the molecule's conductivity in a predictable way.

Sounds neat, huh?

There's only one problem: neither Biernat's nor IBM's memory devices ever saw the commercial light of day. Indeed, IBM has reportedly suspended R&D efforts to perfect a molecular flip-flop because the technology involved was deemed unworkable in practice and (more importantly) not competitive with existing semiconductive technologies.

And that leads me to point number two: Ritchie's offhand reference to "organic chemistry taking care of the whole process" of biochip fabrication hardly does justice to the current theoretical (not to mention practical) uncertainties surrounding molecular electronics generally and biochips particularly. The basic idea behind the biochip concept, you see, is to employ proteins (those big amino acid chains which make up a large percentage of you and me) as the multi-state switching elements in the sort of molecular electronic array I mentioned earlier. One conceivable wrinkle to this scenario would be the use of genetically engineered microorganisms to synthesize and then deposit the desired portions in a predetermined grid or layered structure. Voila—biochip!

Of course, there are some minor questions remaining, you understand, a few details needing clarification.

Like, for instance, we don't have the foggiest idea (well, maybe one or two foggy ideas) how proteins might function as conductors in molecular circuits. Or how they may be put together with the kind of extreme precision that fabricating such circuits would demand. Or how they might be packed together in a circuit having dimensions expressed in nanometers (a nanometer equals about 400 millionths of an inch) without being fried by the heat generated during the circuit's normal operation. Or how. . .

Well, you get the picture, I guess. But, paraphrasing Faraday, nothing is too difficult to be doable. *If* the money's right, that is, and that may be the thorniest problem of all, given the tremendous gains already being made in shrinking down silicon-based chips.

"O.K., Mr. Cynic," you say, "All right. But just suppose, huh, *just suppose?* Fine, says I, let's suppose that biochip technology were sufficiently advanced to permit the construction of a computer from protein molecules. Such a computer would be eensy-weensy, with a capital "E." So small, in fact, that it could fit in eyeglass frames, or even in a human eyeball, or even inside a human skull. A transmog for sure, right?

WRONG.

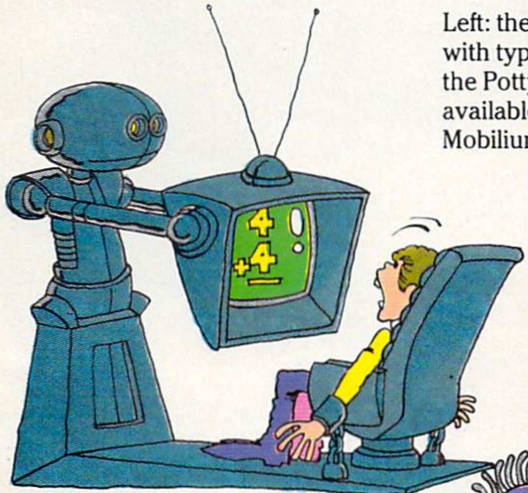
If size were the only hurdle to the hybridization of man and computer, I'd already have my own direct linkup with an IBM 7090 in place. But the real problem is not implantability, or even achieving physiological compatibility between nerve endings and circuit input/outputs. No, the ultimate obstacle to transmogs pertains not to the manmade components themselves, but to interfacing them with a system several orders of magnitude more complex, namely, the human nervous system.

Until we begin really to understand how the components of *that* system communicate with each other (and we're decades away from such insights, I think), talk about connecting the "human mind with the man-made 'mind' of a computer" will be just a lot of smoke.□

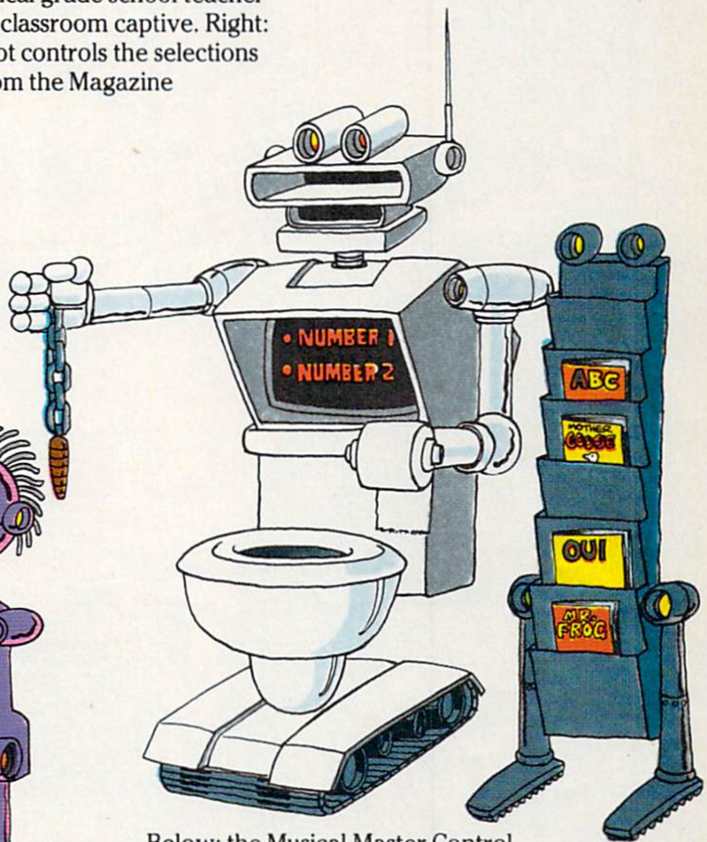
Stephen Bent, a patent attorney who specializes in biotechnology issues, is an associate with the firm of Schwartz, Jeffery, Schwaab, Mack, Blumenthal & Koch, P.C. of Alexandria, Virginia.

Teachers Of The 21st Century!

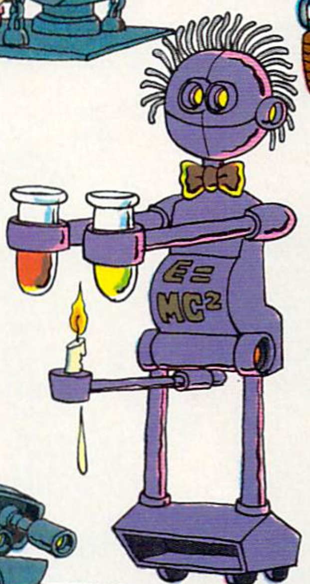
Even five years ago, it seemed improbable that computers would come to play a major role in the education of our children. Likewise, robotics being the crude science that it is today, it seems unlikely that robots will ever serve a useful function in education. But rules were made to be broken, expectations held only to be dashed, and common sense maintained only to be turned topsy-turvy. In that spirit, we project the following developments for our offspring.

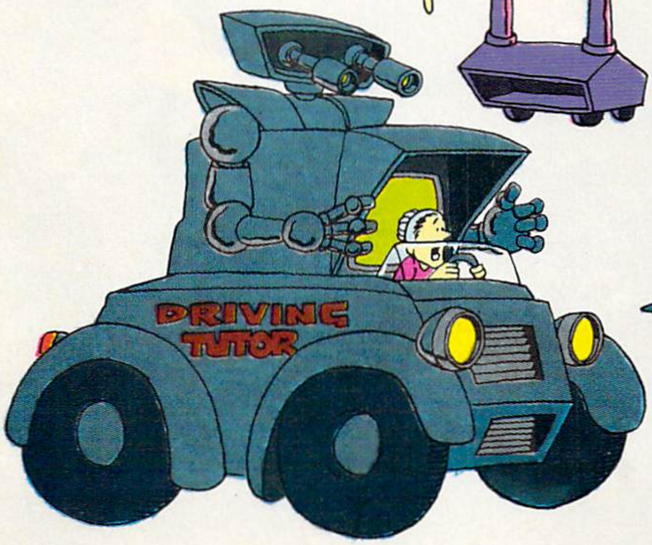


Left: the typical grade school teacher with typical classroom captive. Right: the Potty 'Bot controls the selections available from the Magazine Mobilium.

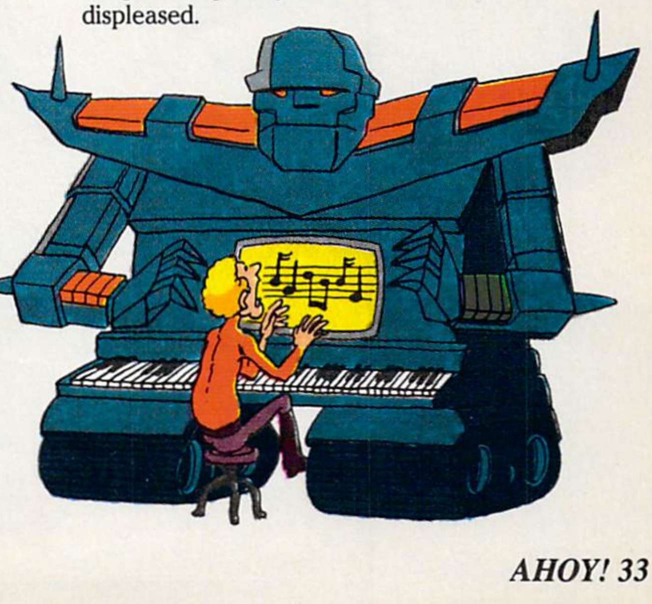


Right: the Chemistry Professor will have an Absent-Minded function built into its program to keep students on their toes. Below: The Captain Bligh Driving Schooner ensures that innocent bystanders need never fear the novice driver.





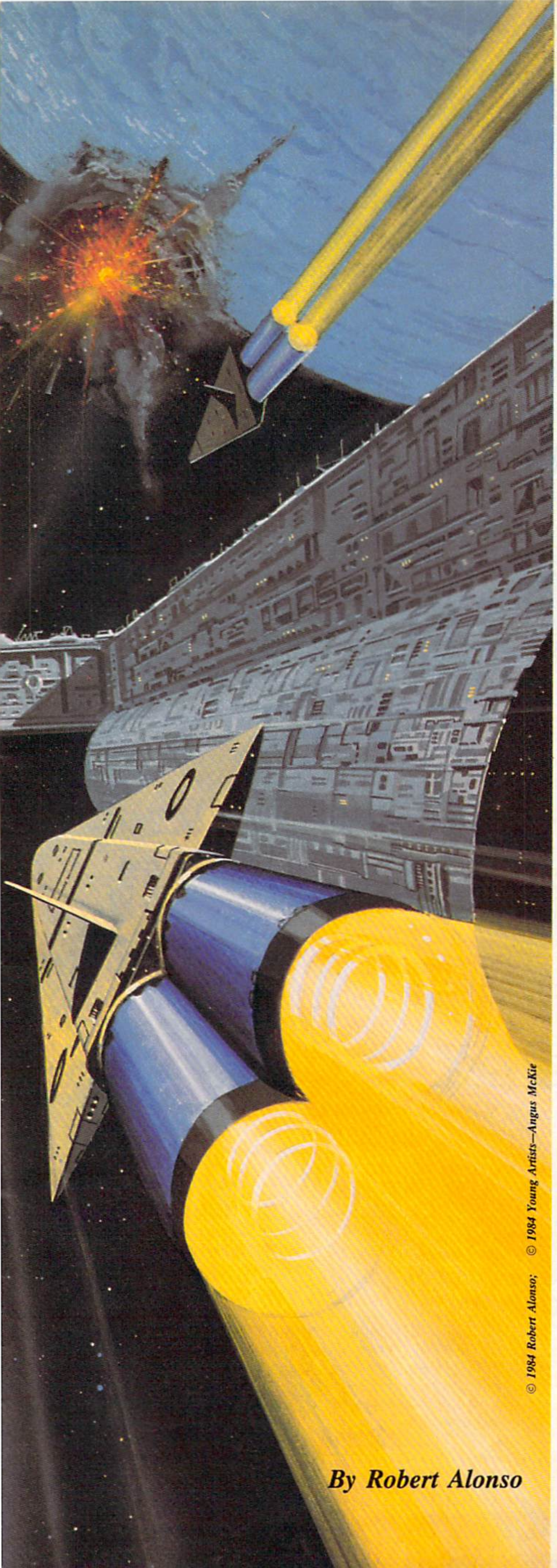
Below: the Musical Master Control Program claps only when it is displeased.





SPACE LANES

FOR THE C-64 AND VIC-20



By Robert Alonso

S*pace Lanes* is a space action maze game with a twist—the action takes place in a maze. You will have to work your spaceship through a dangerous maze to collect fuel dots for the fuel-starved earth. The omnipresent danger is a relentless and random barrage of laser fire from invisible KOPEC robots that patrol the outskirts of the space lanes. To obtain high scores in this game you can only count on agility and strategy; aggression is not permitted. Your spaceship isn't even equipped with weapons.

I've included a VIC-20 and a Commodore 64 version of this game. When typing in the game you will probably notice that the listings for both versions are very similar. This is because both machines have an identical version of BASIC. I tried to structure the games so that equivalent statements could be compared by those of you thinking of upgrading to a Commodore 64. The most obvious differences are screen size and location. The sound memory locations are also located differently. Sound, in fact, is the most incongruent part of programming both machines. The Commodore 64 has a superior sound chip and, I might add, it is more difficult to understand.

Many tricks were used to make the game faster. For example, the main game loop is kept very close to the beginning of the program. This is done because BASIC searches for a line that has been GOSUBed or GOTOed by searching from the beginning of the program until it finds it. It makes obvious sense to place routines that are used most frequently as close to the beginning of BASIC as possible. The laser routine and scoring routine follow the main loop because they are used frequently also. Another trick employed is to assign the moving object (the spaceship) to character location zero, which would normally be the "commercial at" symbol. The reason for this is that the Commodore computers handle a zero expressed as a period (see lines 5 and 40) much faster than they would any other number. Keep in mind that only the zero can be expressed as a period, and only when it isn't part of another number.

The joystick routine in the Commodore 64 version is done with an array. This is done because it is much faster than using eight IF THENs to test for each direction.

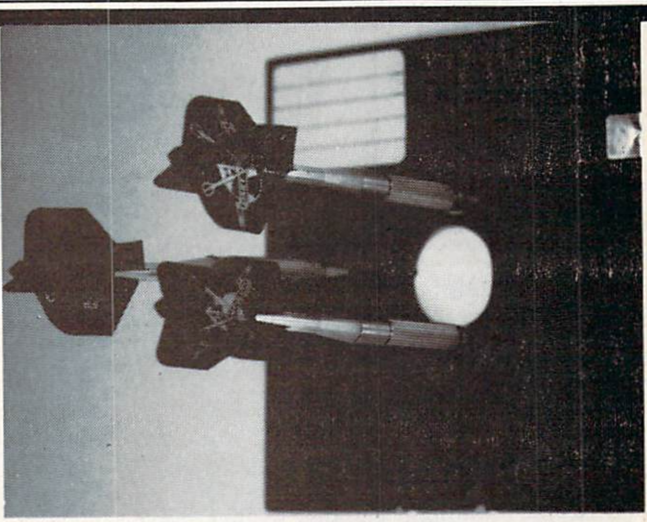
Have fun with these programs and experiment with them. You could try changing the screen color or just about anything else. You'd be surprised how much you'll learn from adapting the program to your taste. If you prefer not to type it in and would like to have it ready to run, send a blank cassette and a self-addressed mailer with $5.00 to:

Nova Soft *(Space Lanes)*
Box 527
Nutley, NJ 07112

Continued on page 76

By Michael Kleinert and David Barron

RANDOM FILES ON THE COMMODORE 64



In the previous two issues we have discussed how to use the more rudimentary file types on your computer—the relative and sequential file types. This month we'll dwell upon a more complex and abstract file type. This is the *random* file.

Both relative and sequential files are machine regulated. By this we mean that the disk drive took care of placing the file name in the directory and placing information in its proper place on the disk. With random files, you must take total control. You are given almost complete command of the disk drive and what it does to your disk. Read the following article with great care, and when experimenting do not use a disk with valuable information. Even the slightest mistake may destroy the contents of a disk.

In order to explain how random files work, we must first talk a bit about how information is stored on a disk. You may have noticed that when you obtain a disk directory, a number appears before each file. This is the number of blocks the file occupies. Each block is a unit of disk memory containing 256 bytes. There are a total of 683 blocks on a disk. Some of these are reserved for the directory and are not available for ordinary use.

The disk is further subdivided into concentric rings called tracks. There are 35 of these on a disk. Each one of these tracks holds a certain number of sectors (each sector is equivalent to a block). Because there is more distance around the outer portions of a disk than the inner portions, more sectors are contained on the outside parts of the disk. Diagram 1 on the following page will help clarify the above concepts.

The following chart indicates how many sectors are contained on each track of the disk:

Track	Sectors
1-17	0-20
18-24	0-18
25-30	0-17
31-35	0-16

You, as a random access file programmer, may want to know what parts of the disk contain what. As we mentioned previously, certain blocks are set aside for the directory. These are all on track 18. As a matter of fact, track 18 is used for nothing else. Since you have direct access to the disk, you need not worry about upsetting the directory, for you won't use it! Random files are not contained on the disk directory. In other words, use whatever parts of the disk your heart desires.

Because of the complexity involved in using random and other files on the same disk, place all random files on a disk not used for anything else. This disk must be formatted since formatting a disk is what creates the tracks and sectors.

Now that we have put the background material to rest, let's get to the programming.

Because of the general nature of random files, we feel that examples will be the best teacher. The following lines of BASIC will write 100 E's to

track 20, sector 5:

```
•10 OPEN15,8,15
•20 OPEN3,8,3,"#"
•30 FORX=1TO100
•40 PRINT#3,"P";
•50 NEXTX
•60 PRINT#15,"B-W:"3;0;20;5
•70 CLOSE3
•80 CLOSE15
•90 END
```

Let's explain this program:

10 Opens command channel to disk drive
20 Opens file for disk data
30 Starts loop
40 Sends a character to the disk buffer
50 Ends loop
60 Writes buffer to disk:

PRINT#15,''B-W:''A;B;C;D
B-W: Stands for ''block write''
A Data channel number from line 20
B Drive number (not device number)
C Track number
D Sector number

70 Closes data channel
80 Closes command channel
90 Terminates program

You may be wondering what the buffer is. This is a place where the disk drive places information before writing it to the disk. First we PRINT all the information into the buffer. When we give it the ''Block Write'' command, the computer writes the buffer onto the disk.

When reading from the disk this process is reversed. First we must bring the information from a specific track/sector into the buffer. Then we read this data from the buffer using GET# or INPUT#. The following program illustrates this:

```
•10 OPEN15,8,15
•20 OPEN3,8,3,"#"
•30 PRINT#15,"U1:"3;0;20;5
•40 FORX=1TO100
•50 GET#3,A$
•60 PRINTA$;
•70 NEXTX
•80 CLOSE3
•90 CLOSE15
•100 END
```

To explain this program:

10 Opens command channel to disk drive
20 Opens file for disk data
30 Reads block from disk into buffer:

PRINT#,''U1:''A;B;C;D
U1: Stands for equivalent of ''block read''
A Data channel number from line 20
B Drive number (not device number)
C Track number
D Sector number

40 Starts loop
50 Reads a character from file
60 Prints character to screen
70 Ends loop
80 Closes data channel
90 Closes command channel
100 Terminates program

The program which follows is similar in nature to the above. It has the added capability of reading from any given track/sector. Once this program has been entered and run, it will show you all 256 bytes of any block you choose. Try viewing track 18 sector 0 on a disk that has some programs on it. Among these odd characters will be the header of the disk. In addition, this block keeps track of where free space is on the disk. Now view track 18 sector 1. This is the first block of the directory. Within it you should find the names of various programs on the disk. If you wish you can view various tracks/sectors and if you're lucky, you may see

Continued on page 76

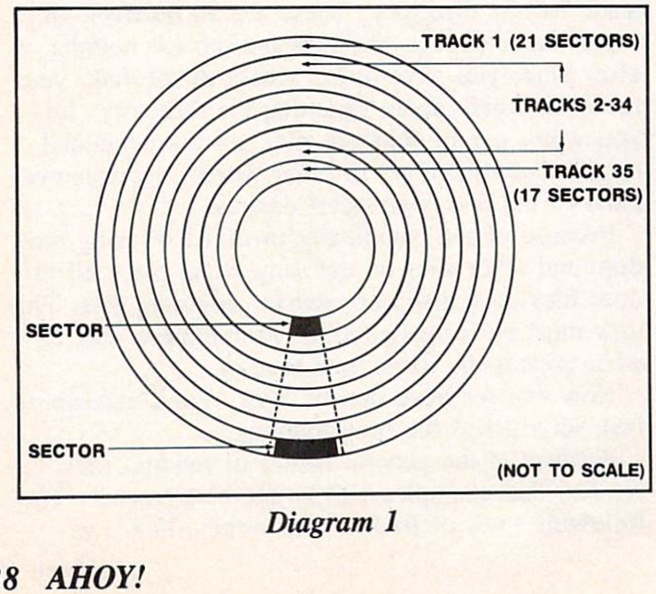


Diagram 1

EASY ACCESS



By B.W. Behling

Have you ever decided to clean out your wallet, only to find two or three mysterious slips of paper with phone numbers on them but no names? Like me, did you then spend the next half-hour looking in your dog-eared old phone book, trying desperately to cross-match names and numbers? If you have, maybe I can help.

What I'm offering is a replacement for that old address book that's not only more efficient, but almost as easy to use. Basically, this is a database management program specifically written to enable you to access your name and address records by entering a minimal amount of information.

USING THE PROGRAM

After typing in the program, save it. You never know for certain that a program is going to work the first time it's booted, and you really don't want to type it twice, do you? (If I had a nickel for every time my cat Ralph knocked out the plug on my computer...) The first thing you're going to be asked when you run the program is your first name. (This is *your* address book, so we want it to be friendly, don't we?) If you're tired from all that typing, just hit RETURN to continue on to the menu.

Now we're at the heart of the program, the menu. As you can see, you have a lot of choices to make. These are all the different files used in the program. You can search your records for a person or company by using any one of these commands. Let's see what they do.

ENTERING DATA

Before you can use any of these commands, you have to have something to search for. So let's look at the ENTER DATA command. Type the letter

"E." The screen will clear and the program will ask for a last name. Let's use yours. Hit RETURN when you're ready to continue. Keep entering the data as it's called for until you get to your phone number. When the program asks for your phone number, you must decide whether or not you want to use area codes. If all your listings are going to be in the same area code you don't need them. However, if you later want to sort things by area code, you'll have to edit all your listings. After you finish your phone number and hit RETURN, the screen will change and you'll be asked to wait while your file is entered. Although this is an honest, hard-working program, it can be a bit absent-minded about who's running it so you'll have to enter your name again. Now the screen should again display the main menu list.

Now you can use the other commands to search for your file. But wait! I told you this was a hard-working program, so let's see *how* hard-working. We'll ask it to find your file by your last name, but we'll only tell it the first three letters. Hit "N," enter those three letters, and press RETURN. Found you, didn't it? You just learned that you don't need to know all of what you're looking for to find a file. Even if you only know the first letter or number of what you want, this will look for anything starting with that character and display it.

EDITING DATA

Now that you know how to enter and display your data, how do you edit it? Easy as pie—maybe easier. First use one of the commands to find the file you want to edit. Then press "E" and the data line (or lines) holding that file will be printed at the bottom of the screen. You are now outside the program, so you can edit the line like any other program line, using the cursor and inst/del keys. When you finish editing press RETURN, bring the cursor down to the bottom, and type RUN. Simple, wasn't it? Now let's talk about how to exit and save the program. Since all of the files are entered into data statements, this program MUST BE SAVED any time new entries are made! If it is not saved you will lose all new files, although any old files will be safe. Saving the program is as easy as typing "X" while the menu is up and following the instructions provided.

IMPORTANT! Hold the shift key down while typing any character underlined (C̲) in the program listing; hold the Commodore key down while typing any character overlined (C̅) in the listing.□

SEE PROGRAM LISTING ON PAGE 83



Educational Software: a guide for parents

PART ONE

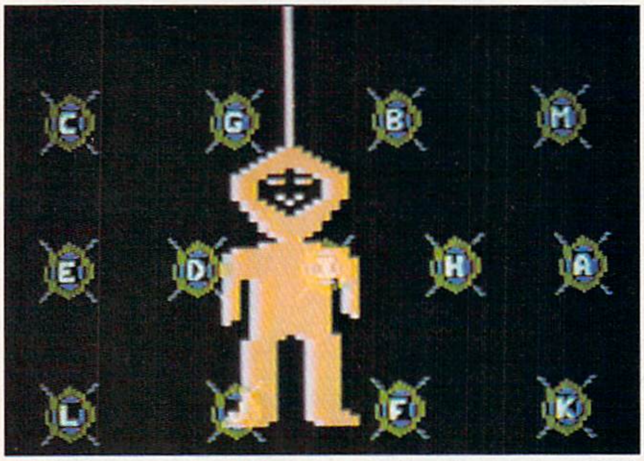
By Richard Herring

Children know which computer programs are fun. Teachers know how to evaluate the educational value of computer programs. Parents feel they should be able to judge both fun and value. But how many times have you as a parent bought, or maybe even written a piece of educational software supposed to be a perfect blend of fun and education, only to have your child look at it once, then walk away bored? Or, even worse, have your child enjoy the "game" so much, become so intent on "playing" it, that you became convinced it must have no educational value?

Is it really possible for a computer game to teach a child, to have meaningful educational goals? Can educational software be fun, motivating and reinforcing the child without imposing learning? The actions of the marketplace seem to say yes. The

Office of Technical Assessment, which advises Congress, estimates that one half of the homes in America will have computers by the end of this decade. Their studies show that, of the families which buy home computers today, almost one half make their purchases specifically to improve their children's educations.

This trend is causing a change in the types and availability of educational software. Just a few years ago, nearly all educational programs involved drill and practice. Although many topics are not amenable to that approach, drill and practice software is the easiest to program. Then there was the initial rash of educational software aimed at the



Alphabet Arcade ***by Program Design, Inc. (a.k.a. PDI).***

home market. Many of the programs were simple. The rationale was that children would be captivated just by the use of the computer, and would not demand the quality programming needed to produce an arcade game or data base manager. Nothing could be further from the truth. Software which can both capture and hold a child's attention and meet some meaningful educational goal must be extremely sophisticated. Only in the last year has the presence of such software really been felt.

The increasing number of home computers has had a direct effect on the type of software available. In the schools, programs were typically designed to be educational with a fun or motivating factor as an overlay. In the home, however, children are unlikely to put up with such programs. Although children are a captive audience at school, they have a wide range of activities available once they get home. The home market virtually demands that educational software be fun first and educational second. When we parents plop down $30 we like to feel that the software will be used more than once without our having to demand that our kids sit down and play. When families are liable to spend as much on educational software for their child in a month as the school does in a year, you can bet that software developers will cater to the demands of parents rather than to those of educators.

Shifting from the school to the home educational market is exactly what some software houses have done. While some people feel that growth in the educational market will cause a surge of buying in the home, other people think just the opposite. Most experts agree, however, that the two markets are mutually exclusive. Spinnaker Software is the leading marketer of educational software for the home. David Seuss, Spinnaker's president, points out that his company directed its emphasis toward home computers primarily because the number of potential sales is so large. Within its programs, Seuss says, Spinnaker tries to produce what the consumers demand. Since a great deal of educational software for the home will be aimed at what sells, a real burden is placed on the consumer.

Parents who buy educational software are not just purchasing a product, they are also casting a vote. That vote says, "This is the type of product that sells. This is a company I'm willing to support." The reason that vote carries so much weight is due to the nature of educational software. When we consumers buy games, we have several sources of information and several ways to cast our votes, but the educational software market is much more confined.

Many home computer and video games are takeoffs on arcade games. The closer they come to duplicating the feel, graphics, and sound of the original, the better. Obvious comparisons to arcade games become our first criterion in purchasing a new home game. There are no arcades where educational software is tested and previewed before it hits the home market. You are not likely to recognize the new educational package on economics, no matter how much fun it is, as a takeoff on that great arcade game you like so much. You can not vote for it with your quarters to help the software company decide that it will sell well in the home market. True, you can look to the school computer lab, but software that works well there may not be appropriate in your home.

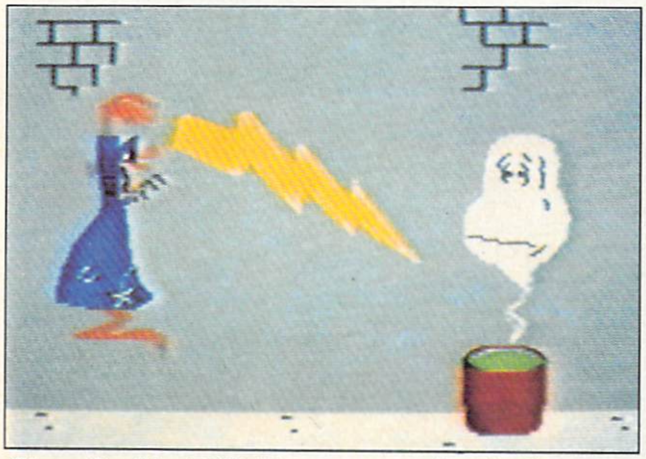
Second, game reviews are a major factor. Just flip through a few pages of this magazine and you will find descriptions that can often tell you whether or not you will enjoy a new game. Or you may find a new strategy that will put life back into play-

ing an old game which seemed forever capable of beating you just before you could get to the next level. Game reviews are easy and fun to read. They describe to us the details of something we already know about—having fun with different types of computer games. But most of us are not experts at evaluating the real teaching value built into software that is labeled as educational. Even after we read the best reviews, we must rely heavily on the comments of the reviewer. We are not nearly as able to evaluate his comments critically or to see his biases by reading between the lines.

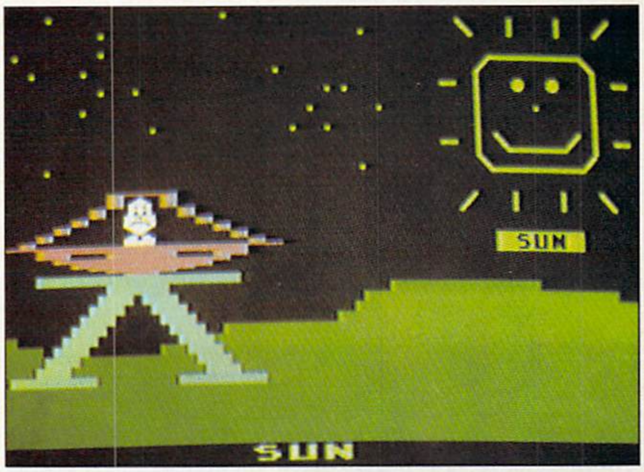
A third source of information for purchasing games is word of mouth. When the family down the street gets the latest release of a hot new game, it won't be long before you hear about it. Excitement over games is not just for kids either. In my town, Chuck E. Cheese at lunchtime looks like a board meeting—replete with three-piece suit types. But educational software is different. No one has ever gotten a sprained wrist from the intensity of playing computer hangman. Sure, your neighbors or your users group will show you new educational software, but it is not as likely to draw a crowd of onlookers as a new game.

That is why the vote you cast when you buy educational software is so important. Your purchase shapes the future of that market much more than when you buy a game. It is up to you to decide what will be available tomorrow. If you shop carefully, you can vote for a world of hits. If you carelessly buy whatever comes along, you will not only waste your money, but also prove to software houses and authors that being labeled educational will sell programs regardless of their value.

Don't get me wrong about educational software being fun either. Nothing says it can't be, or that it



Sierra On-Line's Wiz Type *stars comic strip characters.*



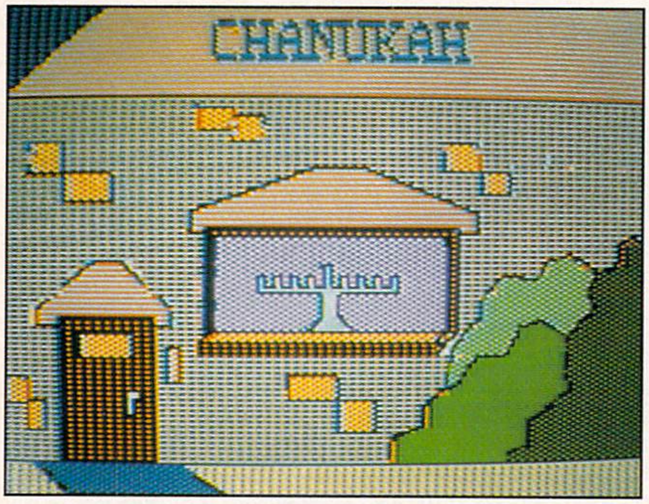
Teddy's Magic Balloon, *another popular PDI offering.*

shouldn't be built around a game. In fact, you are in very good company if you expect educational software to be fun. Dr. Sandra Curtis is the director of research for Joyce Hakansson Associates, a developer and reviewer of educational software. Dr. Curtis maintains that the first goal of educational software is to be fun. The second goal is to teach.

We have all seen the learning that takes place along with play as children watch Sesame Street. Leona Schaubel is director of educational development for the Children's Computer Workshop (affiliated with Sesame Street). Schaubel points out that we aim education at kids based on their age and maturity. She says we must also determine what is fun for each age group and incorporate that in the design of educational software. Sherwin Steffin, founder and chairman of EduWare, describes his DragonWare product line as fun. "We all know that in order to maintain the attention of very young children, and in order for them to acquire some basic core skills, the programs have to be entertaining." EduWare is one of the largest educational and home software publishers.

Young children provide a model for incorporating learning into fun. Or is it fun into learning? Have you ever helped a toddler with ABC's? Did he learn by playing or by furrowing his brow and studying a typeset list of letters? The computer gives us the capacity to carry the marriage of fun and learning to increasingly older students. Even some of the latest tutorials on business software seek to use the capabilities of the computer to encourage learners in ways which are impossible with books or lectures.

The age at which a child's growth can be aided



Davka's **All About Chanukah** *includes an animated dreidel game. (Apple version is pictured.)*

by computers is yet another issue for parents. How old do your kids have to be before they can benefit from educational software? There are programs out there that are suitable for virtually any age child. Very young children soon discover the magic that occurs every time they drop food. It falls. (And adults scurry.) Likewise, they can discover the magic of changing shapes, sizes, and sounds at the simple press of a key (or pound of a keyboard, if your infant's reactions are typical).

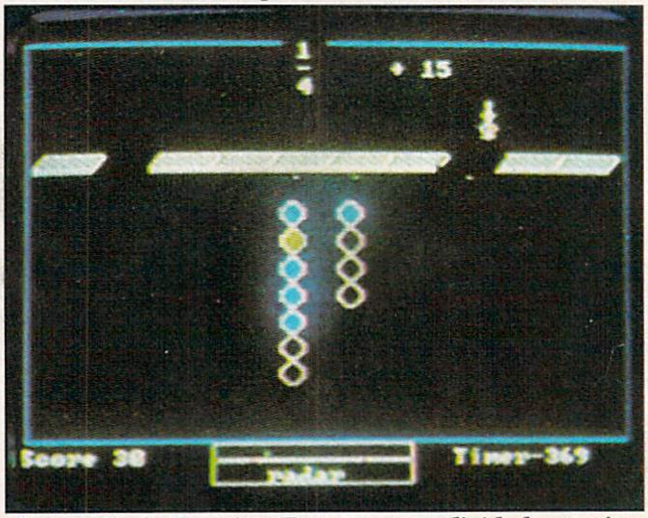
Can it ever be too early to introduce kids to technology which, many claim, will be the center of our lives in the not-so-distant future? You only need to look at the number of bills introduced in Congress during the last two sessions to see how seriously our lawmakers take this new technology. Senator Bill Bradley (Democrat from New Jersey) is one of the members of Congress known as Atari Democrats, a title earned for stressing education aimed at high technology. Bradley says, "If we're moving from a manufacturing-based economy into an information-based economy, then we have to begin to put some emphasis on the teaching of a generation of students." Of his six-year-old daughter Theresa Anne, who completed her first grade introduction to computers, he says, "I wouldn't say right now that she's ready to write her first program. But give her time; after all, she's only six."

Given this complicated picture, how are parents to make wise purchases of educational software for their children? How can you get the most educational bang for your hard-earned buck? Reading software reviews and best seller lists in magazines are certainly ways to start. But shouldering the responsibility of choosing software yourself is the surer method of meeting your child's needs. In the next few issues, we'll discuss guidelines for you to use in selecting programs.

We will find ways for you to set goals for your child's educational activities in your home. Your personal values, as close to or as distant from mine as they may be, will guide some of your decisions. We will consider specifically what you should look for in advertising and packaging. What are the outward signs of a good educational program and what forbodes money wasted on an unused disk or tape?

The program's documentation and the company's warranty policy will both be important to you. There are some simple clues for evaluating these. And since these are programs for kids, you will want to know how to check for error handling and ease of use. We will go into a lot of detail on measuring the educational value of a piece of software, from who wrote it to what the screen shouldn't look like. Along with descriptions of the different approaches educational software can take to helping your child learn, we will explore which approaches are best-suited to different educational goals.

Specific companies and programs will be mentioned occasionally. These references are not meant as absolute endorsements. Rather, they will be examples of some of the outstanding features for which you can look when you are evaluating software. Finally, we will confront the detractors of home computer education. We'll set up their arguments and, one at a time, knock them down. In the end, all this will be rolled together into a checklist that you can use as a tool in deciding which educational software to purchase and which to avoid. □



Spinnaker's **Fraction Fever** *gets undivided attention.*

COMMODARES

By Dale Rupert

Each month, we'll present several programs designed to get the old disk drive spinning—that is, the one located between the ears. We invite you to send your solutions to *Commodares*, c/o *AHOY!*, 45 West 34th Street—Suite 407, New York, NY 10001. We'll print the cleverest, simplest, shortest, or most unusual solutions. Your original programming problems would be equally welcome!

PROBLEM #1: PALINDROMES

Palindromes are sequences which are the same whether written forward or backward, e.g. 12321, otto, madam i'm adam. The user types a word or expression. The computer tells whether it is a palindrome or not.

PROBLEM #2: REVERSALS

The user types a word or expression and the computer types out the reversal of that input, e.g. COMPUTER becomes RETUPMOC.

!KCUL DOOG DNA NUF EVAH

PROBLEM #3: LOGICAL CONCLUSION

```
A AND NOT(B OR C) = A OR NOT D
A OR B OR C = FALSE
WHAT IS THE VALUE OF D?
```

How to determine the value of D?

Hint: there are sixteen possible permutations of ABCD. Eight of them are 0000 0001 0010 0011 0100 0101 0110 0111 where 0 means FALSE and 1 means TRUE (unlike your computer's notation). One way to solve this problem is to have the computer generate a truth table of each of the three terms involving A, B, C, and D.

PROBLEM #4: ILLOGICAL CONCLUSION

```
A) NOT(A OR B) = NOT A AND NOT B
B) NOT(A AND B) = NOT A OR NOT B
C) A OR B = NOT(NOT A AND NOT B)
D) A AND B = NOT(NOT A OR NOT B)
E) A AND (A OR B) = A
F) NOT A AND (A OR B) = A AND B
```

One of these six logical expressions is invalid. Use your computer to find it.

We were delighted with the large number of you who answered our first issue's *Commodares*. Not surprisingly, there were as many different solutions as there were replies. Look, for example, at the answer to *Square Names* supplied by William L. Murphy of Bothell, WA:

```
1 REM AUTOMATICALLY CENTERS
2 REM NAME OR PHRASE ON SCREEN
3 REM
4 REM HIT ANY KEY TO START OVER
5 REM
6 REM TYPE 'E' TO END
7 REM
10 DIM P$(25)
20 PRINT"(SC)
30 PRINT"INPUT NAME OR PHRASE"
40 PRINT:INPUTA$
50 T=((11-(LEN(A$))/2)-1)
60 PRINT"(SC)"
70 FORC=1TOT:PRINT:NEXT
80 PRINTTAB(T+1);A$
90 FORX=1TOLEN(A$)
100 P$(X)=MID$(A$,X,1)
110 NEXT
120 FORX=1TOLEN(A$)
130 PRINTTAB(T);P$(X);TAB(LEN(A$)
+T+1);P$(X)
140 NEXT
150 PRINTTAB(T+1);A$
160 GETE$
170 IFE$=""THEN160
180 IFE$:"E"THENEND
190 RUN
```

which does the job. But Roger Griswold of Palo Alto, CA was able to do the job considerably more easily by eliminating REM statements, using abbreviations, and taking various other shortcuts that you'll want to figure out for yourself:

```
1 INPUT"NAME";B$:L=LEN(B$):PRINT"
"B$:FORX=1TOL:PRINTMID$(B$,X,1)SP
C(L)MID$(B$,X,1):NEXT:PRINT""B$
```

If *Crossed Name* crossed you up, here's the solution spouted by R.C. Fawcett of Kincaid, KS:

```
10 INPUTN$:X=LEN(N$):PRINT"{SC}"
20 FORI=1TO(39-X)STEP2
30 PRINTTAB(I);N$:PRINT"{CU}";TAB
((40-X)-I);N$:NEXT
40 GETA$:IFA$=""THEN40
```

Listed below, is yet another solution for the *Crossed Name* puzzle. This answer comes to us by way of Mark Nekic (Eastlake, OH). As you will notice the main difference between Mark's answer and the one offered by R.C. is that Mark's runs longer (perhaps Mark attended the same programming school as William Murphy). Brevity is the soul of wit, but programming can run to all lengths.

```
10 LC=53272:POKELC,23:DIMROW(25)
20 IFA$="{F1}"THENPRINT"{SC}":POK
ELC,21:END
21 PRINT"{SC}":INPUT"NAME";N$
25 L=LEN(N$):SLOPE=(42-L)/25:AC=5
4272
28 FORX=1TO24:ROW(X)=1523-(INT(X/
2))*41.6:NEXT:ROW(X)=1024
32 INPUT"SIZE (2-25)";NL:Y=40+SLO
PE-L
34 IFNL>25THEN32
36 IFNL-L<1THENPRINT"TOO SMALL":G
OTO32
38 P1=ROW(NL):SP=NL*SLOPE-2:PRINT
"{SC}"
40 P=P1:C=P+AC
50 FORX=1TOL:A=ASC(MID$(N$,X,1))
70 POKEP,A:POKEC,1:C=C+1:P=P+1:NE
XT
80 IFT=1THENIFP>=P1+40*(NL-1)-SPT
HENT=0:GOTO130
100 P=P+Y:C=P+AC
110 IFP<=P1+(40+SLOPE)*(NL-1)+SPT
HEN50
120 P1=P1+SP:Y=40-(41-L)/25-L:T=1
:GOTO40
130 GETA$:IFA$=""THEN130
140 GOTO20
```

At press time, we'd also received correct solutions to various problems from Joe Cook (Paw Paw, MI), Larry Rising (Sonoma, CA), Charley Blair (Mineral Wells, TX), Dex T. Peterson (LeRoy, MI), Larry D. Moody (Houston, TX), Maria Villa (New York, NY), Richard M. Bertke (Perryville, MO) and Gregg Leonard (Dayville, CT). Special honorable mention goes to David Alan Wright (Middletown, CT), whose *Square Name* solution matched Roger Griswold's for brevity, and to Larry Hamrin (Big Pine, CA), whose solution printed the names in different colors.

We'll present more variations next time. Meanwhile, we're looking forward to seeing your solutions to the *Commodares* posed in this issue. □

DYNAMIC POWER FOR YOUR 64!



By Michael Kleinert and David Barron

In this article, we will show you how to enable your computer to "type" things into itself. By this we mean, enter program lines or direct commands just as you would ordinarily type them in from the keyboard! This will allow your programs to actually modify themselves while running. The "dynamic keyboard" technique gives you this power.

Before we do this, we must talk a little bit about what happens when you type something into the computer. As you type on your keyboard, the characters must first pass through something called a "buffer." While the computer is doing other tasks, the keyboard buffer can hold up to nine characters. There are nine bytes of memory in the computer set aside to hold these characters while they are waiting to be processed. In addition, there is one memory location that the computer uses to keep track of how many characters are in this buffer at any time.

The technique we will use involves "fooling" the computer into thinking that someone typed something into the buffer, when we will actually use the POKE command to place the characters into it from within a program.

As you may know, Commodore computers have full screen editing. This means that if you move the cursor over a line of text and hit return, it will be processed. The dynamic keyboard technique involves printing the commands or program lines onto the screen, having the computer type a carriage return over them by placing it into the buffer, and then exiting from the program. We must exit from the program so that the computer will find the characters in the buffer and execute them.

The following is a short program that will add one line to itself, followed by a complete explanation.

```
10 PRINT"{SC}50 PRINT";CHR$(34);"
HELLO, THIS IS LINE 50!";CHR$(34)
20 PRINT"GOTO40"
30 POKE631,19:POKE632,13:POKE633,
13:POKE198,3:END
40 PRINT"LINE 50 HAS BEEN ADDED."
```

Line 10: Prints the line that we wish to add to the program.

Line 20: Prints "GOTO40" directly below the program line so execution is resumed at line 40 once the program line is added.

Line 30: Puts a "home cursor" and two carriage returns in the keyboard buffer and then ENDs the program temporarily.

Line 40: Prints a message on the screen to verify the addition of line 50.

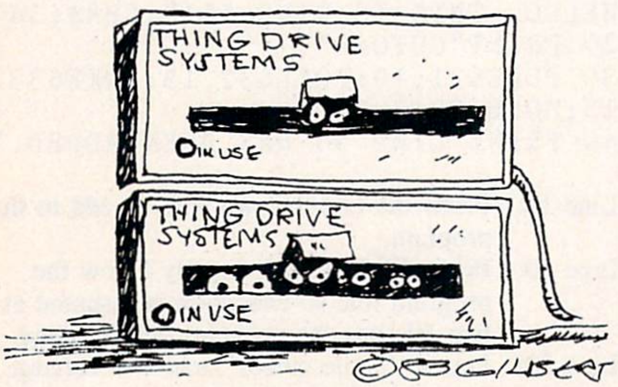
Line 50: Is now part of the original program, and is executed.

Line 30 is a bit confusing, so let's examine it in further detail. Lines 10 and 20 have already placed the program line and GOTO command at the top of the screen. To process these two lines, we must place into the keyboard buffer the CHR$ codes for a home cursor and two carriage returns. The home cursor brings the cursor up to the beginning of line 50 and the first carriage return then enters GOTO40 which continues the program. The keyboard buffer is found at locations 631-640. The first processed character is placed at location 631, and consecutive characters are placed after that. Do not change any locations other than these, as this will cause unpredictable results!

Placing the characters into the buffer is not enough. . .we must tell the computer how many characters are in the buffer by POKEing into location 198. In the above case, we want to process a total of three characters, so we POKE198,3. Once everything has been set up, we exit to BASIC and the characters are processed.

A good application for this technique is the production of DATA statements. Listing 1 contains a program that will do just that. This is useful if you have a large amount of data to add to a program. Listing 1 is an easy-to-follow, fully documented program. Listing 2 is a compacted version of Listing 1 and can be more easily typed in and used with other programs. For this reason we have used line numbers above 60000.

SEE PROGRAM LISTINGS ON PAGE 82



VIC GAME BUYER'S GUIDE

By Walter Salm

If you think the VIC-20 is an out of date, also-ran computer, think again. True, many VIC owners are trading theirs in, giving them to friends or children, or simply putting them on the closet shelf. The reason: the Commodore 64 with its hefty memory and growing library of sophisticated software is the logical tradeup for VIC owners.

But if you've already gotten rid of your VIC, that's a shame, because for all its shortcomings, the VIC is still an excellent machine, and has many exciting features that even now make it more than worthwhile.

If it's still on the shelf, dust it off an plug in some of those cartridges and cassettes that you may have forgotten about. Some of them were outstanding! And they still are. Since there are more than a million of those machines out there, software manufacturers are going to continue to crank out programs and enhancements for this onetime best-seller.

In case you've forgotten, the VIC is not only a great little learning computer, but also a super game machine; and today, it's priced lower than most of those game-only systems. It takes an Atari-type joystick, which means you can enhance your gaming with any one of a number of Atari-compatible joysticks and trackball controllers that are on the market—and these make the gameplay even better.



Slot, *an electronic one-armed bandit, features realistic graphics and casino action—plus an awfully favorable win rate.*

Poker, *also with user-friendly odds, requires you to get a pair of Jacks or better. And it won't fill the room with cigar smoke.*



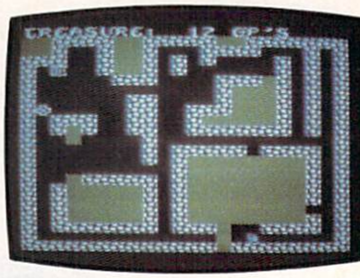
CARTRIDGES GALORE

Back in the early days, the only decent game cartridges came from Commodore itself, probably because only Commodore programmers really understood the rather convoluted memory and screen maps used in their baby. It wasn't long before game cassette tapes and cartridges from "third party" manufacturers appeared. These first efforts were generally pretty bad, because most of these companies didn't have ex-Commodore programmers on staff.

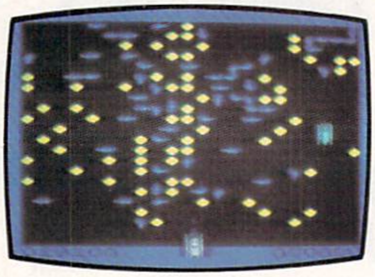
After awhile, as understanding of the VIC's innards grew, good game cartridges started to hit the market from such companies as UMI (regrettably now in Chapter XI), Thorn EMI, Imagic, Creative Software, and (last year) Parker Brothers and Sierra On-Line. Some of these games are excellent; unfortunately, quite a few others are mediocre.

One of the very first game cartridges that I plugged into the VIC when it was spanking new a few centuries ago was Commodore's classic *Jupiter Lander.* I dusted it off the other day, and found it to be as difficult and engrossing as ever. It's still on a par with or better than the majority of current games for any computer.

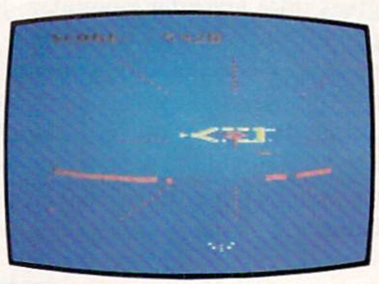
Back when the VIC was the one and only computer in our house, we spent many long nights playing Commodore's two casino cartridge games: *Slot* and *Poker.* The *Slot* game still amazes us with its realistic graphics and action. It's a dyed-in-the-wool Las Vegas or Atlantic City electronic slot machine, and lets you bet up to five coins (for five ways to win). You start with a bankroll of 80 coins, and the win rate is definitely in the player's favor. Even though the win combinations are strictly picked at random (there's no house edge or percentage here), you have to be extraordinarily unlucky to lose at this game.



Sword of Fargoal *lets you go underground to fight off a variety of horrifying adversaries... something like the New York subway.*



Arachnoid, *a UMI game in the* **Centipede** *mode, is harder and not as varied as the same company's* **Video Vermin.**



One of the four scenarios of **Gorf,** *Commodore's VIC adaptation of the challenging arcade game. Don't lose sleep over it!*



The high-speed bowling ball you roll in **Ricochet** *makes a 45-degree turn every time it hits a wall, pin, or bumper.*

The *Poker* game is also like the casino video poker machines. You bet up to nine coins out of your starting stash of 80, and try to get a pair of Jacks or better to win. Again, the odds seem to be stacked in your favor except when you decide to go double-or-nothing at the conclusion of a winning hand. Since we border on being compulsive gamblers in our house, this cartridge too has occasioned many a late night, thankfully without the cost involved in doing the same thing at a casino.

ARCADE-INSPIRED GAMES

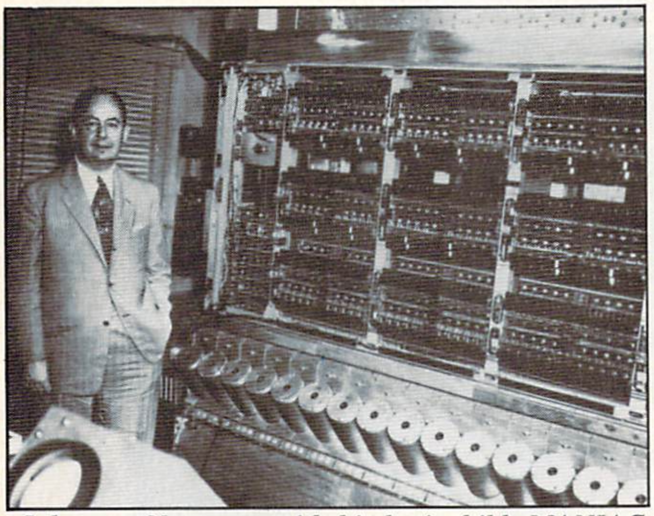
It's sad that UMI (United Microware Industries) is now in bankruptcy. Although the company didn't make it financially, they turned out some dynamite cassettes and cartridges for the VIC. My favorite of the bunch is *Video Vermin,* and if you can come across a copy of this, grab it. Thematically, it's very much like the arcade game *Centipede.* It's an engrossing and exciting twitch game, especially when played with a Wico trackball controller.

Another UMI game cartridge that's quite worthwhile is *Arachnoid*—a later variation on the *Video Vermin* theme. Once again, it's vaguely reminiscent of *Centipede,* but is altered significantly. It's harder and not as varied and exciting as *VV,* but still gets top grades in graphics and playability.

The most significant 1983 VIC entry from our British cousins at Thorn EMI was *Fourth Encounter*—a classic twitch-type shoot-'em-up that is vaguely reminiscent of Activision's *Megamania* for Atari. In many ways, it's much harder than *Mega-*

Continued on page 73

FLOTSAM



John von Neumann with his brainchild, MANIAC.

I enjoyed the article on John Von Neumann, "The Genius Behind Computers," in the first issue of *AHOY!* In addition to more general interest pieces of this nature, I'd be interested in seeing construction project articles in future issues (such as: remote alpha-numeric keypad with function keys onboard; light pen, with software examples; RS232C interface; EPROM burner; etc.).

Also consider letting readers have the option of sending away for a disk or cassette containing all the programs in a given issue. It would sure beat typing in all those programs (not to mention debugging all my typing errors).

Roger Griswold
Palo Alto, CA 94303

We'll continue presenting articles of the types you requested, Roger. As for sparing your typing finger, we're working on a couple of different ways do exactly that. Watch for an announcement!

Concerning the Peanut v. the 64, we have been using Commodore's machine for our business and our farm for six months. It has paid for itself many times over. There are some applications we're looking into that I don't believe the PCjr can be set up for.

A third party relay box now being manufactured allows the C-64 to monitor temperature and turn various appliances on and off. In the near future I expect to be hooking this device into farrowing houses and grain bins to monitor grain and room temperatures so fans and heaters can be turned on and off. The only other way I know of to do this is with a farming system costing several thousand (I think) dollars. My estimated cost will be $400-$500, plus my computer system.

By the end of 1984, an estimated two million C-64's will be in circulation. All those machines in use should spawn more software than you or I could possibly imagine. Applications grow every day!

Allan Wasserman
Chamberlain, SD 57325

(P.S.: your magazine was necessary. You've earned three gold stars.)

Sorry, I cannot accept your invitation to subscribe to *AHOY!* until you add a checksum program to your program listings. Keep in mind that most Commodore owners use a blurry-screened TV as a monitor and have no printer—making it very difficult to proofread a typed-in program.

David Jensen
Clare, MI 48617

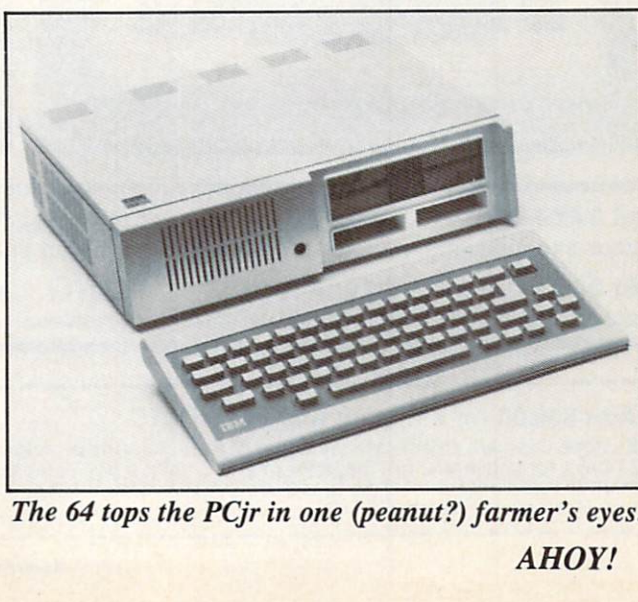
I recently purchased your magazine from a local newsstand. Some of your articles are interesting, and your rating of games available for the Commodore computers is a good idea.

But I would have thought that a checksum proofreader would be of great assistance to those entering listings. Having to go through the entire listing to find that the program does not work is extremely time-consuming and difficult.

Ronald Goldman

We bow to superior wisdom. The team of Michael Kleinert and David Barron is even now working on a proofreading program.

Continued on page 56

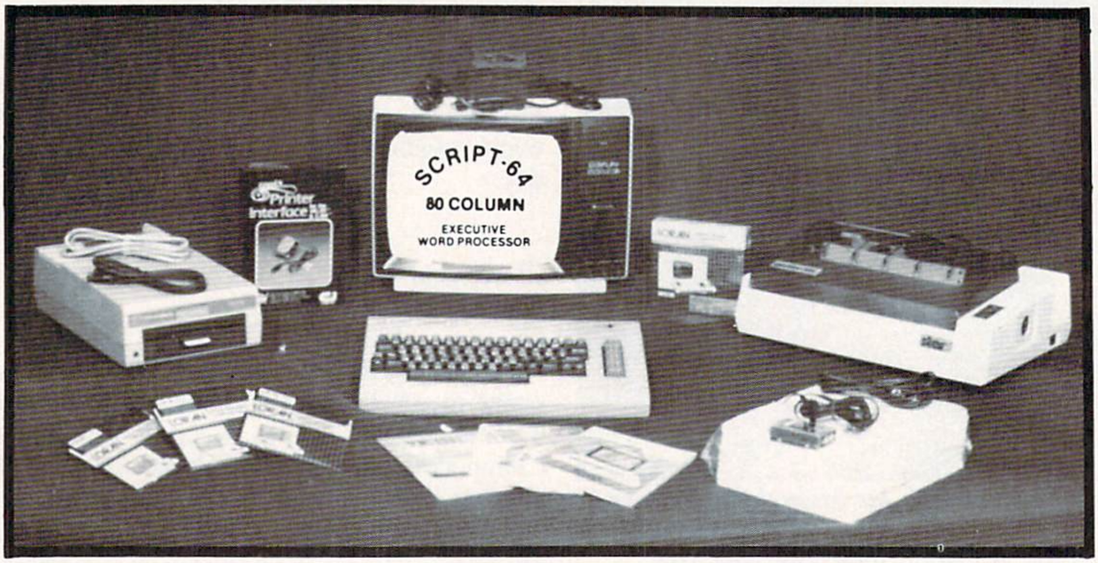


The 64 tops the PCjr in one (peanut?) farmer's eyes.

Complete
WORD PROCESSING SYSTEM $995

(Featuring: "THE SCRIPT 64" 80 COLUMN EXECUTIVE WORD PROCESSOR)

\\ Developed by Richvale Telecommunications Ltd. //



LOOK AT WHAT YOU GET FOR ONLY $995.00

1. **"THE SCRIPT 64"** Executive Word Processor, 80 column in color or black and white, 20,000 word dictionary, data base, mail merge and labels! Developed by **Richvale Telecommunications Ltd.** **2.** The powerful 84K **Commodore 64 Computer!** (more features than Apple II) **3.** 170K **Commodore 64** Disk Drive! **4.** Box of 10 **"Loran"** Disks! **5. Gemini 10X Starmicronics,** 10" Carriage Deluxe, 120 CPS, Dot Bit Addressable, Tractor-Friction Printer!**6.** Deluxe **Cardco** Printer Interface! **7.** Box of Printer Paper! **8.** 12" Green or Amber Screen **"Sanyo"** Monitor! **9.** Monitor Interface Cable! **10. "Special Software Coupon"** that allows $100 to $300 savings!

PLUS! THE **Olympia** OPTION!!



You can add the **Olympia** Electronic Typewriter/Computer Printer for **Superb Letter Quality** correspondence and documents! Now you can have the best **Electronic Typewriter** made and used by the world's largest corporations (better than IBM Selectric) plus — a **Professional Computer Printer** (two machines in one) for only $395.00 additional cost!! List Price $799.00. (Competitive word processing systems sell for $3000 to $5000 without the electronic typewriter feature!!!)

15 DAY FREE TRIAL We give you 15 days to try out these SUPER SYSTEM PACKAGES!! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price!!

90 DAY IMMEDIATE REPLACEMENT WARRANTY If any of the SUPER SYSTEM PACKAGE equipment or programs fail due to faulty workmanship or material we will replace it IMMEDIATELY at no charge!

Add $50.00 for shipping and handling!!

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

Reader Service No. 91

COMMODORE 64

(more power than Apple II at half the price)

$99.50*

- 170K DISK DRIVE $159.00*
- TRACTION FRICTION PRINTER $109.00*

COMPUTER AND SOFTWARE SALE

WE HAVE THE BEST SERVICE

WE HAVE THE LOWEST PRICES

VIC-20

(a real computer at the price of a toy)

$69.50

- 40-80 COLUMN BOARD $59.00
- 32K RAM EXPANDER $95.00

* COMMODORE 64 COMPUTER $99.50

You pay only $199.50 when you order the powerful 84K COMMODORE 64 COMPUTER! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your computer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net computer cost is $99.50!!

SOFTWARE BONUS PACK $29.95

When you buy the Commodore 64 Computer from Protecto Enterprizes you qualify to purchase ONE SOFTWARE BONUS PACK for a special price of $29.95!! Normal price is $49.95 (40 programs on disk or 24 programs on 5 tapes).

* 170K DISK DRIVE $159.00

You pay only $259.00 when you order the 170K Disk Drive! LESS the value of the SPECIAL SOFTWARE COUPON we pack with your disk drive that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied, your net disk drive cost is $159.00.

* TRACTION FRICTION PRINTER $109.00

You pay only $209.00 when you order the Comstar T/F deluxe line printer that prints 8 1/2 x 11 full size, single sheet, roll or fan fold paper, labels etc. 40, 66, 80, 132 columns. Impact dot matrix, bi-directional, 80 CPS. LESS the value of the SPECIAL SOFTWARE COUPON we pack with your printer that allows you to SAVE OVER $100 off software sale prices!! With only $100 of savings applied your net printer cost is only $109.00.

80 COLUMN BOARD $99.00

Now you program 80 COLUMNS on the screen at one time! Converts your Commodore 64 to 80 COLUMNS when you plug in the 80 COLUMN EXPANSION BOARD!! List $199 **SALE $99** PLUS—you also can get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET. List $59.00 **SALE $24.95** if purchased with 80 COLUMN BOARD!! (Tape or Disk)

80 COLUMNS IN COLOR EXECUTIVE WORD PROCESSOR $69.00

This EXECUTIVE WORD PROCESSOR is the finest available for the COMMODORE 64 computer! The ULTIMATE for PROFESSIONAL Word-processing application! DISPLAYS 40 OR 80 COLUMNS IN COLOR or Black and White! Simple to operate, powerful text editing with a 250 WORD DICTIONARY, complete cursor and insert/delete key controls line and paragraph insertion, automatic deletion, centering, margin settings and output to all printers! Includes a powerful mail merge. List $99.00 **SALE $69.00.** 20,000 WORD DICTIONARY - List $24.95 **SALE $19.95.** EXECUTIVE DATA BASE - List $89.00 **SALE $59.00.** (Disk only).

SPECIAL SOFTWARE COUPON

We pack a SPECIAL SOFTWARE COUPON with every COMMODORE 64 COMPUTER-DISK DRIVE-PRINTER-MONITOR we sell! This coupon allows you to SAVE OVER $100 OFF SALE PRICES! $200-$300 savings are possible!! (example)

PROFESSIONAL SOFTWARE COMMODORE 64

Name	List	Sale	Coupon
Executive Word Processor	$99.00	$69.00	$59.00
Executive Data Base	$89.00	$59.00	$46.00
20,000 Word Dictionary	$24.95	$19.95	$14.95
Electronic Spreadsheet	$89.00	$59.00	$46.00
Accounting Pack	$69.00	$49.00	$32.00
Total 5.2 Word Processor			
Tape	$69.00	$56.00	$37.00
Disk	$79.95	$63.00	$42.00
Total Text 2.6 Word Processor			
Tape	$44.95	$39.00	$26.00
Disk	$49.00	$42.00	$29.00
Total Label 2.6			
Tape	$24.95	$18.00	$12.00
Disk	$29.95	$23.00	$15.00
Programmers Helper (Disk)	$59.00	$39.00	$29.95
Basic Tutor (Tape/Disk)	$29.95	$24.95	$15.00
Typing Teacher (Tape/Disk)	$29.95	$24.95	$15.00
Sprite Designer (Disk)	$16.95	$14.95	$10.00
Medicinemen (Tape)	$19.95	$17.95	$12.00
Weather War II (Tape)	$19.95	$17.95	$12.00
Professional Joy Stick	$24.95	$15.95	$11.00
Light Pen	$39.95	$19.95	$16.95
Dust Cover	$ 8.95	$ 6.95	$ 4.60

(See other items in our catalog!)

Write or call for Sample SPECIAL SOFTWARE COUPON!

EXECUTIVE QUALITY PROFESSIONAL BUSINESS SOFTWARE

The Cadillac of business programs for Commodore 64 Computers

Item	List	*SALE
Inventory Management	$99.00	$59.00
Accounts Receivable	$99.00	$59.00
Accounts Payable	$99.00	$59.00
Payroll	$99.00	$59.00
General Ledger	$99.00	$59.00

(*COUPON PRICE $49.00)

VIC-20 COMPUTER $69.50

This 25K VIC-20 computer includes a full size 66 key typewriter keyboard color and graphics keys, upper/lower case, full screen editor, 16K level II microsoft basic, sound and music, real time floating point decimal, self teaching book, connects to any T.V. or monitor!

40-80 COLUMN BOARD $59.00

Now you can get 40 OR 80 COLUMNS on your T.V. or monitor at one time! No more running out of line space for programming and making columns! Just plug in this Expansion Board and you immediately convert your VIC-20 computer to 40 OR 80 COLUMNS!! List $129. **SALE $59.00.** You can also get an 80 COLUMN BOARD WORD PROCESSOR with mail merge, terminal emulator, ELECTRONIC SPREAD SHEET!! List $59.00. **SALE $24.95** if purchased with 80 COLUMN BOARD! (Tape or Disk).

32K RAM EXPANDER $95.00

This cartridge increases programming power over **8 times!!** Expands total memory to 57K (57,000 bytes). Block switches are on outside of cover! Has expansion port!! Lists for $199 **(OUR BEST BUY!)**

60K MEMORY EXPANDER $49.00

Sixslot — Switch selectable — Reset button — Ribbon cable — CARDCO. A must to get the most out of your VIC-20 Computer!

8K RAM CARTRIDGE $39.00

Increases programming power 2 1/2 times. Expands total memory to 33K (33,000 bytes). Memory block switches are on outside of cover! Includes FREE $16.95 game.

16K RAM CARTRIDGE $55.00

Increases programming power 4 times. Expands total memory to 41K (41,000 bytes). Memory block switches are an outside cover! CARDCO Includes FREE $29.95 adventure game!!

12" GREEN SCREEN MONITOR $99.00

Excellent quality GREEN PHOSPHOROUS VIDEO MONITOR with antiglare, 1920 characters (80 characters x 24 rows). Save your TV! a must for 80 column word processors. PLUS $9.95 for VIC 20 or Commodore 64 Cable.

12" AMBER SCREEN MONITOR $119.00

Premium quality AMBER VIDEO MONITOR With antiglare, (80 characters x 24 rows), exceptionally clear screen, faster scanning. PLUS $9.95 for VIC 20 or Commodore 64 Cable.

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY

• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. VISA — MASTER CARD — C.O.D.

PROTECTO ENTERPRIZES

(WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

SANYO MONITOR SALE!!



9" Data Monitor

- 80 Columns × 24 lines
- Green text display
- East to read - no eye strain
- Up front brightness control
- High resolution graphics
- Quick start - no preheating
- Regulated power supply
- Attractive metal cabinet
- UL and FCC approved

- *15 Day Free Trial - 90 Day Immediate Replacement Warranty*

9" Screen - Green Text Display	**$ 79.00**
12" Screen - Green Text Display (anti-reflective screen)	**$ 99.00**
12" Screen - Amber Text Display (anti-reflective screen)	**$119.00**
14" Screen - Color Monitor (national brand)	**$249.00**

Display Monitors From Sanyo

With the need for computing power growing every day, Sanyo has stepped in to meet the demand with a whole new line of low cost, high quality data monitors. Designed for commercial and personal computer use. All models come with an array of features, including up-front brightness and contrast controls. The capacity 5 × 7 dot characters as the input is 24 lines of characters with up to 80 characters per line.

Equally important, all are built with Sanyo's commitment to technological excellence. In the world of Audio/Video, Sanyo is synonymous with reliability and performance. And Sanyo quality is reflected in our reputation. Unlike some suppliers, Sanyo designs, manufactures and tests virtually all the parts that go into our products, from cameras to stereos. That's an assurance not everybody can give you!

SANYO

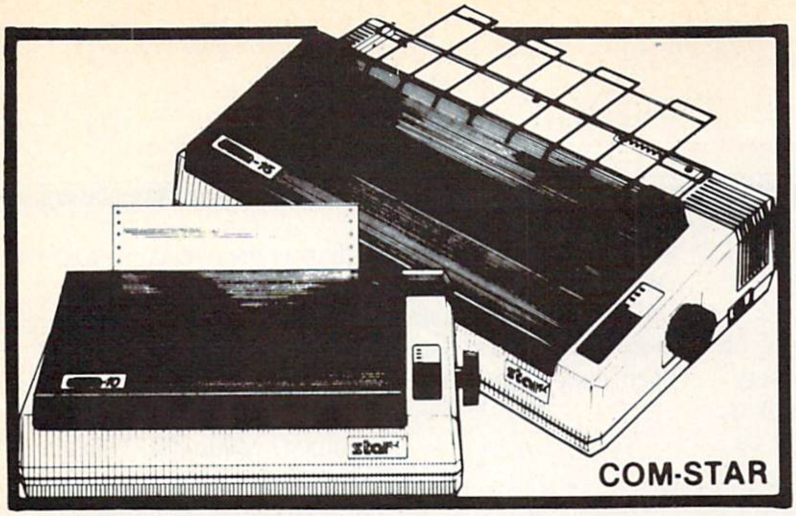
Official Video Products of the Los Angeles 1984 Olympics

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII orders. WE DO NOT EXPORT TO OTHER COUNTRIES.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. Visa - MasterCard - C.O.D.

PROTECTO
ENTERPRIZES (WE LOVE OUR CUSTOMERS)
BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order



FANTASTIC PRINTER SALE

as low as **$149.00**

• *15 Day Free Trial - 180 Day Immediate Replacement Warranty*

	LIST	SALE
80 COLUMN THERMAL PRINTER — 60 CPS Bi-directional, dot matrix, prints 8½″ letter size paper, full 80 columns, high resolution graphics, dot bit addressable, special symbols and true decenders! (Centronics parallel interface)	$199	**$149**
80 COLUMN TRACTOR-FRICTION PRINTER — 80 CPS Bi-directional, dot matrix, impact, prints single sheets, continuous feed paper, adjustable columns, 40 to 132 characters! Roll paper adapter $32.95. (Serial or Centronics parallel interface)	$399	**$209**
PREMIUM QUALITY 10″ CARRIAGE T/F PRINTER — 120 CPS Bi-directional, impact, 9 x 9 dot matrix with double strike for 18 x 18 dot matrix. High resolution bit image (120 x 144 dot matrix) underlining back spacing, left and right margin settings, true lower decenders, with super and sub scripts. Prints standard, italic, block graphics, special characters, plus 24 of user definable characters and much more!! Prints single sheets, continuous feed and roll paper! (Centronics parallel interface)	$499	**$289**
PREMIUM QUALITY 15½″ CARRIAGE PRINTER — 120 CPS Has all the features of the Premium Quality 10″ Carriage T/F Printer above plus a 15½″ carriage and more powerful electronic components to handle large business forms! (Centronics parallel interface)	$599	**$379**
HIGH SPEED PREMIUM QUALITY T/F 10″ PRINTER — 160 CPS Save printing time with these plus features: 160 CPS speed, 100% duty cycle, 8K buffer diverse character fonts special symbols and true decenders, vertical and horizontal tabs. This is Red Hot Efficiency!!! (Serial or Centronics parallel interface)	$699	**$499**
HIGH SPEED PREMIUM QUALITY T/F 15½″ PRINTER — 160 CPS Has all the features of the 10″ Carriage high speed printer plus a 15½″ carriage and more powerful electronics to handle larger business forms! (Serial or Centronics parallel interface)	$799	**$599**

PARALLEL PRINTER INTERFACES: (IN STOCK)

- For VIC-20 and COMMODORE 64 — **$49.00**
- For all APPLE COMPUTERS — **$69.00**
- For ATARI 400 and 800 COMPUTERS — **$79.00**

NOTE: *Other printer interfaces are available at computer stores!*

SALE SALE

WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail! Canada orders must be in U.S. dollars. We accept Visa and MasterCard. We ship C.O.D.

Reader Service No. 94

PROTECTO ENTERPRIZES (WE LOVE OUR CUSTOMERS)

BOX 550, BARRINGTON, ILLINOIS 60010
Phone 312/382-5244 to order

FLOTSAM

Continued from page 51

Congratulations to you on your premier issue of *AHOY!* I'd like to say that it was perfect in every way. That is what I would like to say, but as it is, I do have a few suggestions for you: They are as follows.

How about a series on assembly language for wide-eyed innocents (guess who) who don't have four years of computer science under their belts. We novices are tired of being treated like lepers by the initiated. Any music utilities are also appreciated, as are any programs which simplify life or enhance its quality.

I don't know what the portrait of an *AHOY!* buyer is, as I'm probably atypical, but one thing you have to realize is that not all computer owners are capable of field stripping a C-64. Me, for instance. I'm a musician. I don't have a four year degree in comp. sci., so when the corners of the mouths of experienced hackers start to curl up into a sneer, I toss them a home-scored version of *Trois Gymnopazdizs,* point to one of my guitars, and tell them to show me how *easy* it is!

I seem to be back to that point again, but it is important. If those "in the know" continue to act as if the beginner is in some way mentally deficient, the beginner will lose interest in the computer, and you lose a potential customer/subscriber! So it's to everyone's advantage to treat the newcomer as a valued asset, rather than a thing of the arcane lore. Agree?

What we really need is a "score & save" music utility, one that offers full bass and treble clefs, staffs, edit facilities, and *not* only 2/2, 2/4, 3/4, and 4/4 times! I transcribe piano music for guitar, and in some music the time signature may change from measure to measure. For instance, in one tune, you start off in 4/4, but before you reach the end of the tune, you have gone through 6/8, 12/8, 5/6, 12/16, and 18/16! *None* of the available music utilities can handle complex time, and I'm sure that many musicians would be very happy if you can come up with one that would.

So that's it: the first issue's great. I can only hope that you continue with the excellent programs contained in it. I'll buy the next couple of issues, and if it looks like you're really publishing user-friendly programs, then, gentlemen, you will have a lifelong friend and subscriber.

Best wishes of the season to you, and one caution: never mix anything with a good single malt scotch, including water. It harms the whiskey, and causes homicidal rage in nearby Scots (like myself). Putting anything in rye is acceptable, since there's not much else you can do to rye that hasn't been done in its manufacture. Wäs Hazl!

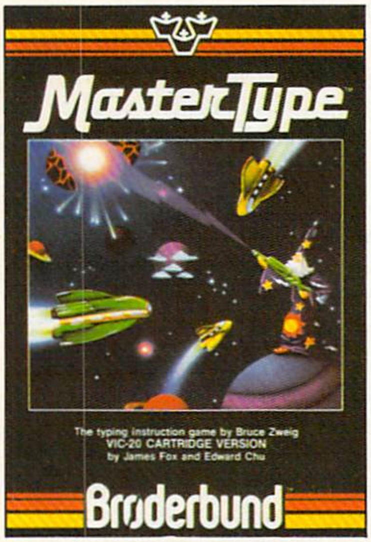
John McMenemy
Winnipeg, Manitoba, Canada

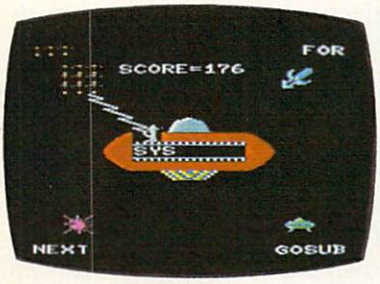
Thank you, John, for the many nice compliments. As far as publishing user-friendly programs, that is our aim. As far as music utility programs that allow for multiple times, we'd love to have them. If any of our readers know of such a program, or have written one, please let us know. In addition, we'll continue to report on and review new musical programs in the coming months.

So what do you think about this issue? For that matter, what do you think about anything at all? Write to Flotsam, c/o AHOY!, 45 West 34th Street—Suite 407, New York, NY 10001.



REVIEWS





MASTER TYPE
Broderbund Software
VIC-20
Cartridge; keyboard
(Disk drive or datasette recommended)

Behold—an arcade-style game that teaches a crucial lesson for computer enthusiasts. *Master Type* is in the pivot-and-shoot class of game; you control a flying saucer that is fixed in the center of the screen while enemy ships attack from all corners. Ah, but there's a rub or two: each enemy ship is represented by a letter (or keyboard symbol at higher levels) and you must depress the corresponding letter or symbol key to dissolve the ship.

At the beginner's mode even I, a hunt-and-peck man, achieved a fairly high score. This is where that all-important storage device comes in. The real fun starts when you can store forty words of up to eight letters or characters each. You'll quickly learn to type without looking at the keyboard or you'll disappear from the game. If you mistype any part of the word the rockets keep coming, the mistake appearing in the window of the saucer. The graphics, incidentally, are wonderful, more lavish than many games of this type, so to speak.

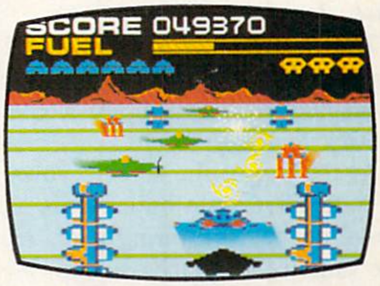
The game can be played with more than one player, with each player finding the most difficult inputs for the other player.

If you have a disk or datasette device, this game is great fun and very useful. If you don't have either device, the game will only take you so far.

Broderbund Software, 17 Paul Drive, San Rafael, CA 94903.

—Steve Stone





BUCK ROGERS—PLANET OF ZOOM
Sega
C-64, VIC-20
Cartridge; joystick, keyboard

In case you've never vacationed there, the planet Zoom is one long ski run. The player, as Buck Rogers, is challenged to navigate through deadly electron posts while battling alien saucers, space hoppers, and—ultimately and inevitably—the alien mothership. There are five levels of difficulty, each with four waves of enemies. Some rounds take place on the surface of sunny Zoom, others in the void of space. The player is allotted three fighterships, with bonus points given for every 20,000 points scored.

Graphics and gameplay combine to make this an excellent and absorbing—though admittedly one-note—space game. The graphics are of a fairly new type, one that has not yet worn out its welcome: rather than the trench warfare of the arcade game, the electron posts appear scrolling from the horizon toward the player on an otherwise flat and arid plane. With the surface scrolling toward the player, a sense of place and of movement is achieved in one stroke.

There is some limited strategy, though the player will have to rely on instinct to effect it. If you play too carefully, fly too slowly, you risk running out of fuel. If you fly outside the posts to avoid that hazard, you risk running into another set just out of current screen range. Concentrate too hard on the enemies before you, and one will sneak up from behind and zap you.

Gameplay is the same in both versions, though I preferred the

VIC version for two reasons: control of the ship was more sure (there is also more space between the posts) and the graphics are actually better. The problem with the 64's graphics—not a crucial one—is the low contast of colors on the planet's surface (you don't get the full sense of onrushing movement) and the sketchy quality of Buck's craft.

Sega, 5555 Melrose Avenue, Los Angeles, CA 90038.

—Dan Hallassey

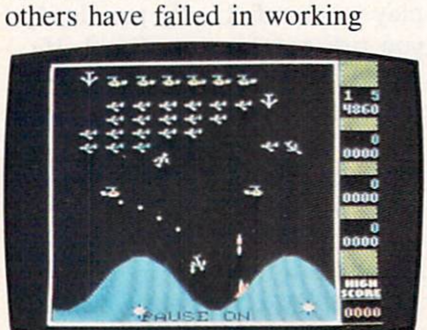
WAVY NAVY

Sirius Software
C-64
Disk; joystick

The object of *Wavy Navy* is simple: kill or be killed. A game description is also simple: it's seasick *Space Invaders.*

At the top of the screen, rows of helicopters rain bullet hails down on your seafaring ship. At the same time, planes swoop down and drop bombs; failing that, they will streak at your ship kamikaze-style and attempt to dispatch you in that fashion. In addition, there are a second type of bomber, minuteman rockets, and underwater mines. But the kicker is this: the waves crest and roll, hoisting your ship from screen top to screen bottom and making accurate defense a nightmare. *Space Invaders* this ain't!

Though I can't recommend the game to everybody, I must admit that Sirius has succeeded where others have failed in working new wrinkles into the slide-and-shoot format. The game has all the features you expect from Sirius: pause button, multiple players, multiple levels, etc. You can even become *President* if you successfully plow through all nine levels.



Sirius Software, 10364 Rockingham Drive, Sacramento, CA 95827.

—Pete Lobl



ENCHANTER

Infocom
C-64
Disk; keyboard

In Infocom's latest prose adventure, the player assumes the role of a young sorcerer of meager ability. Armed only with a spell book, the apprentice is charged by the Circle of Enchanters to penetrate the gloomy keep of Krill the Warlock and put an end to that villain's festering reign of terror.

On the road and within the environs of Krill's castle, the player must make the acquaintance of all manner of creatures and humans and gather scrolls; these contain spells which must be transferred to the spell book and memorized. The player must have a myriad of spells in store before the confrontation with Krill takes place. He's powerful and clever and all that evil sorcerer type stuff.

Any more detail would of course give too much of the game away. Ah, what's one little hint? When you first enter the gate of the castle, if you take a right or left you will find yourself in total darkness with a "puissant" beast that would love to "eat your person." Use the frotz (light) spell, but not on the room of the beast. The spell won't work and your person will get et. Cast the spell on yourself and sally on.

Enchanter is filled with the usual Infocom doses of wit, red herrings ("Burma Shave!?"), twists, turns, and surprises. The time limit is not a problem. Just be sure you are well-fed and hydrated. Beginners will be delighted with the wealth of vocabulary at their disposal, but more experienced *Zork*-ers and *Suspended* addicts may be disappointed. There didn't seem to be a whole lot to do. I solved the game in record time, for me; it left me wanting more. Good showmanship, bad gamemanship?

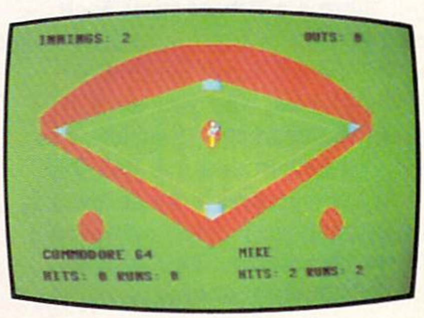
What there was of Krill's castle provided suspense (especially if you know you only have one more command or else you die—you'd better get it right) and wonder. The very moniker "Infocom" may be raising my expectations too high.

Infocom Inc., 55 Wheeler Street, Cambridge, MA 02138.

—Pete Lobl

BIBLE BASEBALL

Davka Corporation
C-64
Disk; keyboard



The Davka Corporation is a software company that specializes in Biblical games and tutorials. They have been producing material primarily for the Jewish community, but now I understand that they are branching out into the Gentile market as well.

Bible Baseball allows the player the option of playing with either computer or human. The game has two levels of play, minor league and major league. Each level has one hundred multiple choice questions that have been gleaned from what Christians refer to as the Old Testament portion of the scripture.

Questions appear in the on-deck circle, Basically, when you play the computer, you take turns, a correct answer or wrong answer determining if a batter is out or has eked out a hit. In the two-player version, the player whose team is at bat fields all the questions, right or wrong answers determining hits or outs. A hit could be anything from a single to a home run, depending on the difficulty of the question.

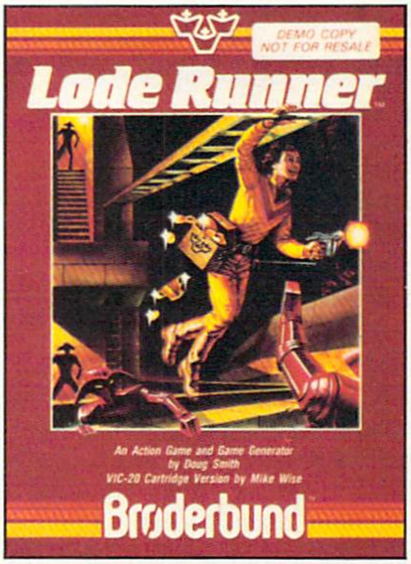
There are nice touches of baseball ambience throughout the game: on-deck batters running out to congratulate home-run hitters; the appearance of the American Flag and the playing of the National Anthem (with appropriate reminders to stand); pitchers being sent to the showers; a game being called on account of rain (a tarp is pulled over the field). Also, a hitter can be robbed. An outfielder will sometimes appear and catch a hit ball (normally, only the infield is shown).

The questions range from names and relationships to events and locations. Not all of them are serious either. Witness: "Samson lost his strength because Delilah... a) Cut his hair, b) Overcooked his food, c) Didn't feed him."

The only real complaint I have about this game is the very limited quantity of questions. With only one hundred questions available in each level, it's likely that players will know all the answers by the third game. But I understand that Davka has made available a supplemental disk with which the player can create his own questions, either from the Bible or any other source. It seems to me that supplemental disk is crucial to long-range enjoyment of the contest (both disks together retail for around $35—the customary toll for quality software).

Davka, 845 N. Michigan Ave., Suite 843, Chicago, IL 60611

—R.J. Michaels



LODE RUNNER

Broderbund Software
C-64
Disk; joystick
(Also available for VIC-20)

Some games require precise timing, quick thinking, and a skillful hand on the joystick. Others call for faultless reasoning and a sharp, analytical mind. To master Broderbund's *Lode Runner,* you will need all these things, and more.

The idea is simple: maneuver your Galactic Commando around a brick structure while collecting gold chests and avoiding the Bungeling guards. What isn't simple is the way in which many of the game's 150 screens are designed.

Often, the gold chests are buried deep within solid walls of brick and must be laboriously dug out via the action button. Other times, chests are perched on inaccessible platforms at the top of the screen and you must trick enemy guards into bringing the gold into reachable territory. Still other screens require you to dig your way into smaller substructures, retrieve the gold, and escape before the bricks seal back up again.

Unlike most climbing contests in which a fall will cost you a game life, *Lode Runner* allows your commando to plummet safely from any distance. Rather than jumping over enemies, your on-screen surrogate can simply dig a hole to escape through, or trap enemies in. Although this takes a little getting used to at first, before long you'll be digging your way out of seemingly dead-end situations.

In addition to being a top-notch action game that requires both a quick mind and an agile joystick, *Lode Runner* is an easy-to-use game generator that allows you to design and save up to 150 of your own custom game screens.

After entering the "Edit Mode" from the game disk, you "Initialize" a formatted diskette with a few simple commands. Once your diskette is initialized, you design new game screens by moving a cursor around and inserting various game elements

such as bricks, ladder section, trap doors, enemy guards, gold chests, etc. Each new item is chosen by typing a number from one to nine. Zero is used to erase.

Once your masterpiece is complete, it is saved with the command CTRL-S. You may then quit the edit mode (CTRL-Q) and playtest the screen. If you want to make additional changes, simply reenter the edit mode. Also, all user-created screens can be moved from one level to another or wiped out completely with a few keystrokes.

With all this going for it, *Lode Runner* is one of the best games available for the C-64. Unconditionally and wholeheartedly recommended.

—Lloyd Davies



CONGO BONGO

Sega
C-64, VIC-20
Cartridge; joystick, keyboard

Your honor, I object. Neither in the game nor in the instructions booklet is this poor gorilla charged with any crime, yet the object for the player is to "get" him.

If that were the only problem with *Congo Bongo,* Sega would be in clover. Alas, the game is fraught with problems: gameplay is repetitive, frustrating, tedious, inconsistent, and at times confusing, and the music not only got on my nerves but stomped on them. Plus, the whole thing is derivative.

There are two screens in *Congo Bongo* and five levels of difficulty. In screen one, the player must climb Jungle Mountain, cross a chasm and a river and get the gorilla while avoiding thrown coconuts and hopping monkeys. In screen two, the player must ford a river by hopping lily pads, hippos and islands. Once crossed, the player must avoid rhinos and other hazards to get the ape. The game then resumes at screen one at a heightened level of difficulty.

This falls into the "cute" or "cartoon" genre of games. If you have any other satisfying entries of that ilk, *Congo Bongo* can be avoided. The game is difficult, and that's fine, but after awhile I found it tedious to climb that damned mountain again. The game is capricious, and that's okay too: hazards you passed easily once will bedevil you on subsequent passes. The game has humor; I especially enjoyed the gyrations I had to undergo to shake off the monkeys. Challenging, surprising and humorous, yes, but not enjoyable, at least for me. One more beef: often, you will think you've avoided a hazard (the visual cue tells you so) but still you die. Give me clarity or give me death.

The graphics are very well done in both versions, differing in the manner that the main characters are rendered. The 64 version is slightly faster.

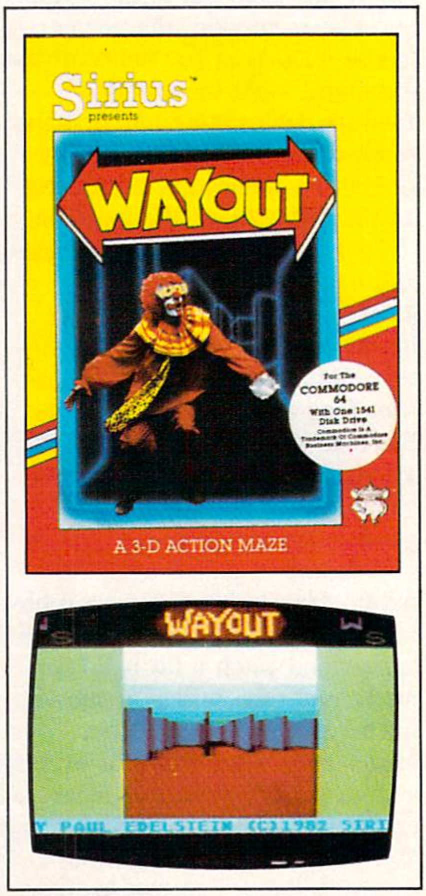
Sega, 5555 Melrose Avenue, Los Angeles, CA 90038.

—Dan Hallassey

WAYOUT

Sirius Software
C-64
Disk; joystick, keyboard

I approached this game with anxiety and a certain lack of qualifications. Maze games generally leave me cold. To my surprise, I found *Wayout* a complete delight, to the point where I



had to be dragged forcibly from the computer.

This is of the new school of maze games: the player is provided with a sketchy overhead map of his or her position in a continually evolving maze, but the main portion of the screen is taken up with the first person point of view from inside the maze. The object is simple: find the pulsing door that indicates the way out, and escape. Complications ensue with the Cleptangle, a spinning rectangle that wanders the maze with you. The Cleptangle will steal your mapmaker (the overhead view will not function) and your compass. A further difficulty is the Wind, which will make certain maze portions impassable.

There are 26 mazes from which the player can choose. Pause and save features are included.

I have seen other games of this type (CBS' *Tunnel Runner* and Starpath's *Escape from the Mindmaster*), and I much prefer *Wayout*. The object here is simple, keeping the frustration level at a minium. Negotiating the maze with the joystick is a pleasure. Changing direction is easy, controlling speed is no problem, and if you run painlessly into a wall it is a simple matter to slide off it and continue. Most maze portions are not constructed of strict right angles—a superb design innovation that makes bouncing around inside a leisure activity rather than an endurance test. The sense of movement and place rank as nothing less than superb.

Maze game fans—pounce on this one.

Sirius Software, 10364 Rockingham Drive, Sacramento, CA 95827.

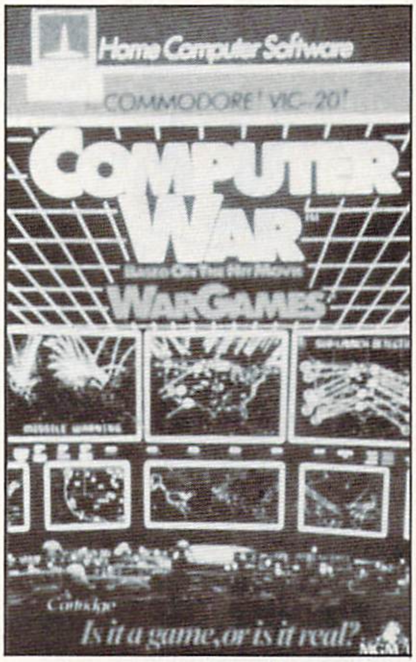
—Dan Hallassey

COMPUTER WAR

Thorn EMI Software
VIC-20
Cartridge; joystick

A warning is issued before you play *Computer War;* the only way to win is not to play. And they're not kidding! You know you're in trouble when it takes 20 minutes to read the instructions, and two more days to fully understand them.

Actually, the scenario is very simple. Someone has penetrated the National Defense Computer and activated a war game. Unfortunately, the computer doesn't know it's a game. Therefore you must destroy every missile while at the same time convincing the computer that it's all a mistake. If the computer is destroyed, or too many missile bases are wiped out, a real missile attack is launched and our green granite planet falls apart.



The first way to lose. Basically, you start with a map of the North American continent with missile bases superimposed. Missiles, represented by little white dots (and an occasional unexplained black one) appear and begin moving towards the bases. You must move the cursor towards a dot and press the fire button. The screen then changes to a terrain display with arrows at the top corners directing you towards a missile. Then you blow it up. Of course, all this is done within a certain time limit. Failure to dispatch the missile results in the loss of that base and a lowering of the defense status. Four such losses and the world is gone.

The second way to lose. While you're engaged in trying to protect your missile bases, a single hit on your computer center immediately ends the game. Obviously, you should make defense of the computer priority number one.

The third way to lose. After catching up to and destroying that elusive missile, you must match a color pattern of random colored squares that the computer selects. This is the part of the game that offers the most trouble. It is also the deadliest. If you fail to solve the puzzle, you almost invariably bring on immediate nuclear holocaust.

Why you can't win. The game goes on endlessly, picking up speed. There is no way to increase your chances of surviving mistakes, and all you can do is get more points than everyone else. That doesn't mean much when the world ends.

Does all this sound familiar? Does all this sound like fun? You bet!

—Steve Stone

DTL-BASIC 64

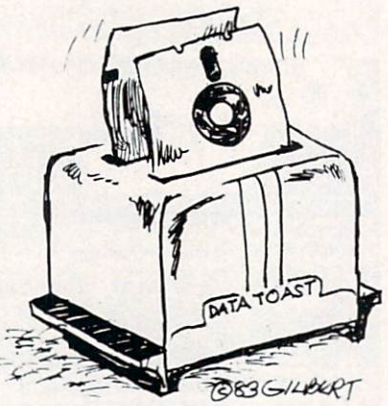
Drive Technology Ltd.
C-64
Disk

It seems that more and more

fine software products for the Commodore 64 are coming from overseas. This is not surprising, as the PET computers achieved considerable popularity in the European market well before the advent of the VIC-20 and the 64. *DTL-BASIC 64,* a product of Drive Technology Ltd., England, is a compiler for the full Commodore 64 BASIC. This means that any program written in BASIC for the C-64 can be compiled. It will handle extensions to BASIC, such as *Simons' BASIC,* as well. (Yes, I did successfully compile and execute a small *Simons' BASIC* program. This was rather surprising as part of the *Simons' BASIC* program, which was in cartridge form, is located under the C-64 BASIC ROM. This same address space is occupied by the compiler run time library in RAM (more on this later). Thus we had the situation where three programs were successfully manipulated by the Commodore 64's sophisticated bank switching capabilities while executing from the same address space.)

Getting back to the subject at hand, the purpose of a BASIC compiler is to convert code written in BASIC to machine language, which should result in higher speed and efficiency. This is accomplished by executing the compiled code directly without having to use the BASIC interpreter. The penalty is that once compiled, a program cannot be easily modified. Any changes must be made on the original source code which then has to be recompiled. This last process is highly disk interactive and does take several minutes, depending on the length of the original program. Most compiled programs will be significantly smaller as well.

DTL-BASIC is well-designed and easy to use. The 45-page manual in an oversized looseleaf binder, although concisely written, is easy to follow. The first chapters are designed to allow a beginner to work with the compiler without using its more advanced features. These features, covered in the later chapters, are for more experienced programmers. Both disk and tape versions are available, as well as a protector version for commercial software houses. The latter version produces output for use with a security key for copy protecting commercial software. The tape



version lacks some of the capabilities of the disk version and is not as convenient to use.

To obtain an estimate of execution speed improvement by the compiler, a short BASIC program, which calculates all prime numbers less than 1000, was compiled. The program, included in the listings section of this issue (see page 83), is a sieve of Erastosthenes' implementation for calculating prime numbers. The BASIC version executed in 26.7 seconds when run under the Commodore interpreter. An unoptimized compiled version ran in 13.1 seconds. When the integer mode compiler directive was used, running time was reduced to 6.8 seconds.

This brings us to the most important features of the compiler, the use of directives. These are special commands, included in REM statements, which give the compiler specific instructions for performing the program compilation. One of the most significant of these directives is the ability to do true integer math as opposed to floating point math. It seems that although Commodore BASIC supports integer variables, all calculations are performed using floating point math, a time-consuming process. This may not be significant for a few calculations, but when it is repeated many thousands of times in a loop the differences really add up. The DTL Compiler provides its own integer arithmetic routines, which are much faster than the interpreter's floating point routines.

In addition, the compiler provides its own garbage collection routines. Garbage collection is the process of reorganizing string storage to recover unused space. This procedure is performed periodically by the BASIC interpreter and can consume some tens of seconds under certain conditions. This slow garbage collection is actually a characteristic of the BASIC 2 used by the Commodore 64, VIC-20, and early PET computers. The upgraded PETs, with BASIC 4.0, use an improved garbage collection routine which is much faster. So are the routines supplied by the compiler. They utilize RAM located in the $C000 block. This means that machine language programs such as the DOS Wedge which live in this area cannot normally be used. The compiler allows the garbage collection pointers to be adjusted to reclaim this area for machine language. If the garbage

collection routines run out of space, the interpreter routines are automatically utilized.

Compiled programs are loaded and run just like ordinary BASIC programs, although a LIST command only shows a single line with an SYS command. An 8 kilobyte run time library must be loaded into the computer the first time a compiled program is run. This is taken care of automatically if the run time library program is copied to the same disk as the compiled program. Since the library routines are loaded into RAM under the BASIC ROM in the Commodore 64, no program space is lost. There will actually be more RAM available for most compiled programs than for the uncompiled versions. As long as the computer is not turned off or reset, additional compiled programs may be loaded and run without reloading the run time library.

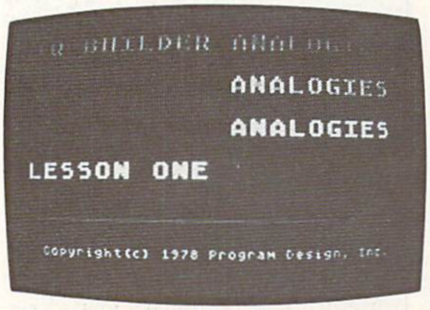
Compilation is a two-phase process. On the first pass, the source file is read a line at a time and a semicompiled version of the program is written to a work file. On the second pass, the work file is read back, additional information is added, and the object file is created. For the tape version, the work file is held in memory, which limits its size to 12 kilobytes. For the disk based version, program size is limited to any size which will run under the interpreter, since the entire program is never in memory during compilation.

Extensive error checking is performed during both passes of the compiler. Error messages are generated and can be listed to screen or printer. Thus an additional application for the compiler exists: that of syntax-checking BASIC programs even if they are not to be compiled.

The program is copy protected by the use of a security key which plugs into the cassette port. This security key must be present or the program will not function properly. Backup copies can be made of the original program disk or tape, and full instructions are included for this purpose. The security key is not needed to run programs which have been compiled by *DTL-BASIC.* Drive Technology includes a release to allow users to distribute programs compiled by *DTL-BASIC* provided mention is made of the compiler's origin. *DTL-BASIC 64* is distributed by Cimarron Corp., Division of Standun Controls, Inc., 2158 Hathaway Street, Santa Ana, CA 92705. Phone: (714)662-2801. Price is less than $100.

—Morton A. Kevelson

SEE LISTING ON PAGE 83



ANALOGIES
PDI
C-64
Disk

"No games!" I announced, lugging the long-awaited computer into the house. "This is not a toy, and we are not going to become the neighborhood video arcade." Lesson One in how to burst a kid's bubble.

Well, I managed to stick to my guns for only three months. (Two and a half, actually, but who's counting except the kids?) Because frankly, last year's outright dearth of educational software for the Commodore 64 drove me first up the wall, then straight to the closest videogame outlet. The keyboard was getting cobwebs!

Fortunately, things are looking up. There is suddenly a wide variety of educational software for your Commodore that'll put smiles on moms' faces. (Unfortunately, though, maybe not on kids' faces.)

To wit: Program Design's new series of programs which include vocabulary builders, analogy quizzes, word recognition skills, spelling tests, you name it! PDI's *Analogies* program is nothing less than a teaching aid. Suitable grades include 6th through 12th. Its applications are numerous. Do you want to practice for the SAT's? It's perfect! Want to improve your English skills? PDI's *Analogies* to the rescue!

The program is set up on 6 escalating levels, which when followed sequentially truly sharpen the student's awareness of analogical usage. Levels 1 and 2 present 60 simple analogies. Example: mouth is to head as

- a) type to one of its characteristics
- b) members of the same class
- c) things with a feature in common
- d) part to whole
- e) group to member

When you answer correctly the computer responds with a little fanfare. If incorrectly, it gives you the correct answer, in this case "d."

As you move on to Levels 3, 4, and 5, things toughen up a bit. Here the task is to complete 50 analogies. Example: painting is to frame as

- a) photo to album
- b) frame to house
- c) bristles to brush

d) water to rain
e) statue to chisel

Not so easy? Wait! The program now offers the choice of a helpful hint, which in this case is: "holds art." Thus the answer is "a."

At Level 6 the program presents 15 analogies in the following pattern: a : b :: c : d. English translation: antiseptic : bacteria ::

a) army : nation
b) lawyer : defense
c) prescription : cure
d) bald : hair
e) education : ignorance

Once again, the hint helps, indicating that "a" removes "b." So—AHA! "e" is the answer.

Got it mastered? Then you're ready for the final test, which includes 16 analogies set up similarly to Level 6. It was at this part of the program, however, when I became disenchanted. Having listed my answers on a piece of paper, I found that the answer sheet at the end of the test only indicated correct and incorrect guesses. If I had guessed wrong, the monitor so indicated, but did not list the correct answer. This method is probably employed to encourage the user to retake the test, but I found it a big inconvenience. I like to know the answer! It's often difficult to figure out for yourself exactly why your answer is wrong unless you have a better answer to compare it to.

Of its ilk and for its intended use, PDI's *Analogies* is a worthwhile program. It would be more fun, and probably more enticing to kids, if it included some graphics or music (anything!) to liven things up. Unless sitting and reading page after page of nothing but analogies is your idea of a really good time.

—Valerie B. Tamis

NUKEWAR
Avalon Hill
C-64 and VIC-20
Cassette

If you're a real game player from way back, the name Avalon Hill is certainly familiar to you. The Avalon Hill Game Company has produced over 150 strategy board games. They publish games of sports, politics, adventure, role-playing, and science fiction, but are best-known for their realistic wargames using maps with hexagonal grids and dozens (or hundreds) or playing pieces.

Continuing their move into the



computer age, the company has produced *Nukewar,* a wargame simulation in which the computer controls your opponent country's strategy. The computer also does the tedious chores: keeping track of allowed moves, determining odds of events occurring, and scoring. Although *Nukewar* is not nearly as involved or sophisticated as most of the non-computerized board games, it is a revealing hint of the kind of games that might be available in the near future. The C-64 version is reviewed; the VIC-20 version is on the same cassette (as are versions for the TRS-80 and Atari 400/800).

Play consists of two phases, Cold War and Hot War. On the screen are two square eight-by-eight grids representing your country and your opponent's. Some of the cells in your grid contain cities, missile bases, and bomber bases, while others are empty for future expansion. Your opponent's map initially reveals nothing.

During the Cold War phase (beginning in 1956 or so) you have the option of spying or building up your armament. Spying reveals some of the details on your opponent's map—essential information when the Hot War begins. During each year interval, you may add to your numbers of missile and bomber bases. After 1965 you may build nuclear submarine bases, and after 1970 anti-ballistic missiles.

At any time, either you or your opponent may declare Hot War. Then the action begins. Whoever declares war has the "first strike" advantage, but world opinion weighs against that player in the final determination of the "winner" of the nuclear war.

During Hot War, you take turns activating your bases. You specify the numbers of planes to be equipped as bombers. You may give coordinates on your enemy's map as targets for your ICBMs. Missiles may drift off-course, so there is no guarantee that your target will be destroyed. Your relative numbers of arms and your knowledge of the enemy's map determine your success.

After a few years of Hot War, the Premier of the opponent country contacts you to negotiate a settlement. If you both agree, a truce is declared. A body count is taken. Whoever has the largest population left is the winner, unless world opinion says otherwise, or unless so few survivors remain on both sides that there is

no winner.

Nukewar is in certain respects an unconventional type of game. For one thing, unlike most arcade games, it doesn't give you a chance to shoot at everything that is attacking. At some points during Hot War, you'll watch helplessly as barrages of missiles wipe out your cities and bases, and there's nothing you can do. The odds of the enemy getting through your lines are determined by your strategy of buildup during the Cold War.

While faster-paced than most of the board/strategy games, *Nukewar* is certainly not a real-time, shoot-'em-before-they-shoot-you action game. More thought and planning are required to win this one (as well as some luck).

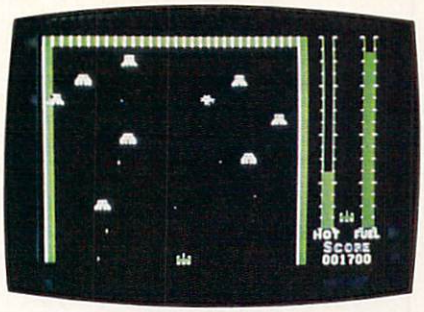
If you want a game that takes more than fast reflexes, *Nukewar* will fill the bill. Though a long way from the sophisticated, multi-hour board games you may be familiar with, it provides a glimpse of some of the things computergames will do in the future.

—Dale Rupert

THRESHOLD
Sierra On-Line
C-64
Disk; joystick

Only humor saves this game from being a run of the mill slide-and-shoot, burn-the-alien-invaders game. It is slightly reminiscent of Activision's *Megamania.*

The player pilots a ship that remains at the bottom of a screen while various waves of aliens pass overhead in horizontal and vertical erratic patterns. Of course they bombard the player's ship with laser fire and the player fires back. But indiscriminate return fire will be penalized, as the player's gun can become overheated and useless for a time.

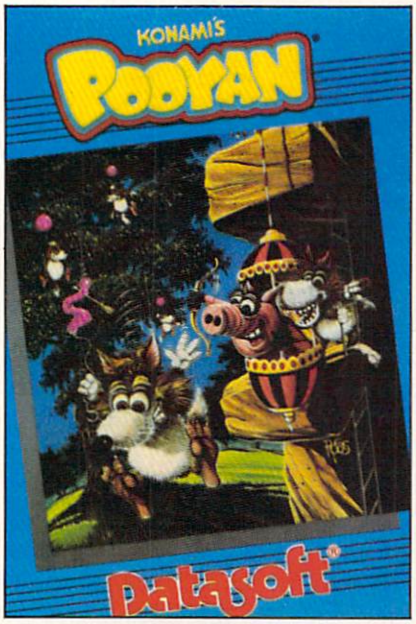


In* Threshold, *you face a host of weird foes. (Apple version shown.)

The aliens appear in the form of oscillating snowflakes, swimming goldfish, chomping mouths and more. The player's Mothership, appropriately labeled "Mother" in huge letters, refuels the player's ship and receives a "Thanks, Ma" message. Once Ma departs, the player is thrust into laserfire chaos once more.

I enjoyed this game to the utmost, though I'm fully aware that, without its humor, it's as derivative and familiar as next week's TV cop show.

—R.J. Michaels



KONAMI'S POOYAN
Datasoft Inc.
C-64
Disk; joystick

Pooyan can be described as a vertical slide and shoot. In two screens—Pooyan's home and the wolves' lair—the object is to shoot arrows at a pack of attacking wolves in air baloons before they can bite you or drop a boulder on you (you're a pig). In the second round, there are balloons within balloons, making this task a bit more difficult. Also, you as the plucky pig Pooyan can toss a slab of meat which will distract the voracious wolves and send them plunging down to their deaths.

The graphics in *Pooyan* are gorgeous and detailed, from the landscape to the movement of the adversaries. Gameplay, however, is very very *very* simple. Although I have seen adults chomping at the bit to get a chance to play this game, I recommend it only for very young children, with this proviso: the graphics being so, well, *graphic,* it may trouble some parents to see little pigs plunging to their deaths and wolves being impaled on cute little arrows.

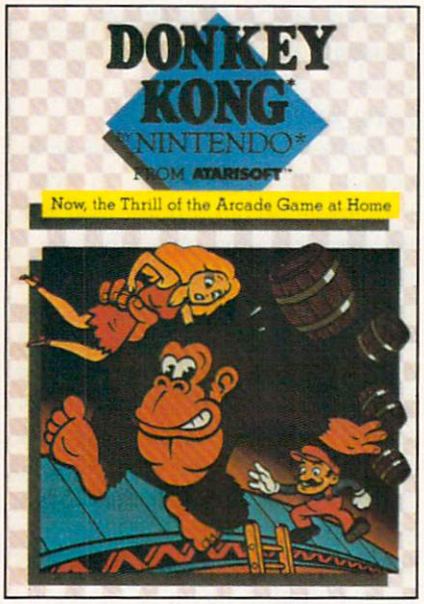
Datasoft, Inc., 9421 Winnetka Ave., Chatsworth, CA 91311.

—Robert J. Sodaro

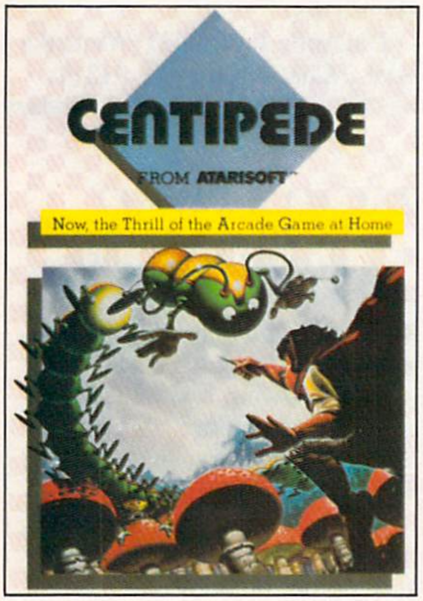
PAC-MAN, DONKEY KONG, CENTIPEDE, ROBOTRON: 2084
AtariSoft
C-64, VIC-20
Cartridge; joystick

The folowing review is based on the Commodore 64 versions of the popular arcade/dedicated home video games.

For those who have been in TV-less hermitage for the past few years:



Donkey Kong is a multilevel, multiscreen climb-and-shoot game, the object being to rescue a fair maiden from a gorilla while avoiding all manner of hazards. *Centipede* is a slide-and-shoot (with limited vertical movement as well) in which the player blasts, segment by segment, a descending centipede and other creepy-crawlies through a grid of mushrooms. *Pac-Man* is a maze game in which the player gobbles fruit while avoiding ghosts and blinkies. Perhaps the least familiar is *Robotron,* a free-form run, shoot, and rescue game. While blasting a bewildering swarm of robots, hulks, mindless grunts, sinister spheroids and their deadly spawn, enforcer embryos, galvanizing electrodes, cubic quarks, and torturing tanks, the player must rescue man, woman, and child.



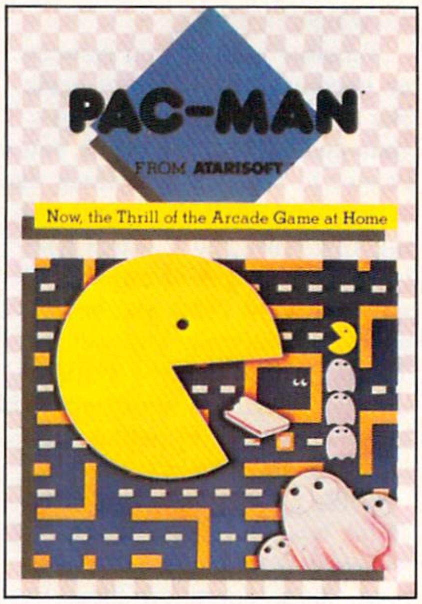
Because most of these games are so familiar, players probably already know whether they want them or not, whether the games will enhance their collection or not, whether a purchase is justified if the games are already owned in dedicated game machine versions. In general, however, Atari's AtariSoft division has done a good—in one case, spectacular—job of adapting the games to the Commodore computers.

Robotron is somewhat reminiscent of *Shamus,* but without the maze motif, overall objective, and wit. Control is unwieldy and the sound effects are decidedly unfuturistic: the laser shoot-'em-up sounds like volleys in an indoor tennis court. But if you like to scramble about and shoot amidst utter confusion, this game is for you. Agreeably chaotic, complicated and yet simple-minded.

Pac-Man is a delight and faithful to the arcade game, though of necessity the gamefield is wider than it is tall. Well-paced. Fine graphics.

Centipede suffers from control comparisons with the arcade's rollerball. Control is not exact, nor are the colors as lustrous as we're used to, but gameplay is simple to learn and is uncannily addictive. Appealing jungle-gurgle sound effects.

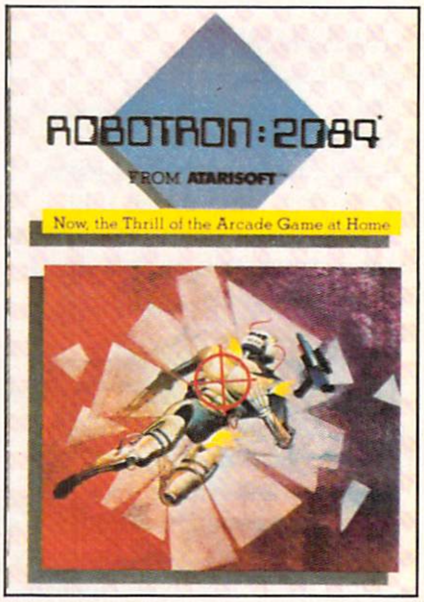
Donkey Kong is the best adaptation of the lot, a complete success from the cute attract mode



to the final dizzying screens. I'm not a *Kong* connoisseur but I think I endured some hazards I hadn't seen before. Graphics are crisp, joystick response is true, and best of all (superior in this respect to the ColecoVision version) the music is a non-irritant. If you've been waiting for a Commodore version of *Kong,* you'll be delighted.

Atari Inc., 1265 Borregas Avenue, P.O. Box 427, Sunnyvale, CA 94086.

—Tim Moriarty



SUPER-TEXT PROFESSIONAL
Muse Software
C-64
Disk

"But how can you do serious word processing with only 40 columns?" my Apple- and IBM-using friends once scoffed. I'd always balked at adding an 80-column board—the question of software compatibility had plagued me. But now a classic Apple word processor with an 80-column display built into the software has been converted to run on the Commodore 64.

Super-Text Professional replaces the 64's standard characters with its own set in order to facilitate an 80-column display. You'll need a hi-res monitor for maximum readability while entering and editing text in 80 columns. Without such a monitor, you can enter your text in the 40-column mode, then switch to the 80-column display (it takes about a minute to load from the program disk, and doesn't destroy your current file) and view it as it will appear on paper.

This means you can check out the appearance of files that employ numerous tab stops, things such as outlines or columns of figures and names. If your work consists of a series of paragraphs, you may prefer to enter and edit in the 40-column mode, whose larger letters are easier on the eyes. It's convenient to have both options readily available.

But even if you *never* used its 80-column mode, *Super-Text* is loaded with features that will increase your efficiency at the keyboard. Onscreen formatting can be activated by pressing a single key. This means that indented paragraphs, centered lines and other formatted effects will immediately be displayed as they will on paper. Automatic word-wrap ensures that any words too long to fit on an onscreen line will drop down to the next line, upgrading readability.

The screen isn't "live," so you can't type in words and edit them in the same mode. "Add" mode is used for keying in text or inserting more copy within an existing file. You enter that by hitting the "f3" key. Tap "f5" for "Change" mode to strike over characters and replace them with others. Pressing "f5" or "f3" again puts you back in "Cursor" mode. If all this seems inconvenient, it at least paves the way for more flexible implementation of a number of powerful and sophisticated functions.

"The Key" is a good example. It's quite simple to define the £ (British pound sign) key to represent up to 30 characters. (I often harness it to type in "powerful and sophisticated features"—that phrase then appears onscreen with the touch of a key, and also on paper when it's time to print.)

From Cursor mode, you also select whether you will print on continuous paper, single sheet or to the screen. Block operations are called up here too, after marking the block to be manipulated with a Control-V at its beginning and end (the block isn't highlighted, though). You can move, copy, delete or unmark a block of any length.

Scrolling is handled uniquely. In Cursor mode, you type a plus or minus, which appears at bottom-screen to indicate whether you wish to scroll forwards or backwards. Then you may hit "L" or "P" to scroll a line or screen at a time. (You can also move to the top and bottom of the file, to the left or right of any line, and to the top or bottom of a screen. Find and Replace is executed in a two-step procedure from Cursor mode. You can replace all occurrences of a word (or several different ones) or answer a prompt to change only the ones you choose. *Super-Text* executes this in the direction your scrolling is set for, so you can Find and Replace backwards, a rare capability. Typing "J" in this mode moves you to the last correction you made.

A key consideration with any word processor is printer compatibility. *Super-Text* can load a "printer parameter" file that lets you customize all settings for any printer on the market, dot-matrix or letter quality. Eight user-definable control keys allow you to utilize special characters or functions of your printer, though it may take a day or so to deci-

pher your printer's manual. Screen colors can be fiddled with until you've got just the right combination of purple letters on a yellow background.

Files can be linked so you can print manuscripts too lengthy to fit into RAM at one time. Most word processors put you through some strange changes if you want to print a single page, or just part of one, from a file. The simplest shortcut is usually to delete everything else in the file—but with *Super-Text* you simply insert a few characters to mark the section you need printed. Everything else will be ignored. And whenever you're knocking out hard copy, the text is simultaneously printed to screen, unlike with most word processors for the C-64.

The split-screen facility is handy. It slices the screen in half horizontally, showing the same text in each. But the top half is active, so you can rewrite part of it and compare it with the bottom half to see which reads more smoothly. All other editing and scrolling capabilities can be used in the top half.

Files are saved by pressing "f7" and typing in the name. Resaving an edited version of a file already on disk is done by keying in a period and leaning on return. "F1" brings up the "retrieve file" menu, and all files in the disk's directory are numbered—you can load one by typing in that number instead of the entire name. All the standard features you'd expect from any word processor—format lines, tabs, automatic page numbering—are included. The only thing missing is the ability to generate form letters (an optional module is available).

And if you think that mastering the many powerful and sophisticated functions of *Super-Text* will take forever, think again. While using 80 columns, just press "H" (in Cursor mode) and a series of four explicit Help screens come to your rescue.

Super-Text Professional is available for $99.95 from Muse Software, 247 North Charles St., Baltimore, MD 21201. Phone is (301) 659-7212.

—Shay Addams

Reader Service No. 87

READING COMPREHENSION: WHAT'S DIFFERENT
PDI
C-64
Disk, cassette

I can get my 12-year-old son to take out the garbage. And he'll even make his bed occasionally. But short of the firing squad, I honestly don't think there's any way I'll convince him to leap to the computer to use PDI's *Reading Comprehension: What's Different.*

Which is too bad because this is worthwhile educational software, and would truly help build the analytical skills necessary for better reading comprehension. But believe me, it's one dull program.

There are ten different lessons, each progressively more rigorous. Level 1 is suggested for grade 2 and lower; Levels 2 and 3 for grade 3; and so it goes until the final 3 levelsd (8,9,10) which are appropriate for grade 6.

In each lesson there are between 35 and 50 examples in which the object is to select the word that is different. Here's a Level 1 example:

father
tree
daddy
man

If you guess incorrectly, the computer moans, and you can guess again. When you've answered properly the computer makes a small cheerful beep, and the monitor flashes the reason you're correct; in this case it's "tree" because a tree is not a male human being.

By the higher levels, the word selection intensifies. Here's an example from Level 8:

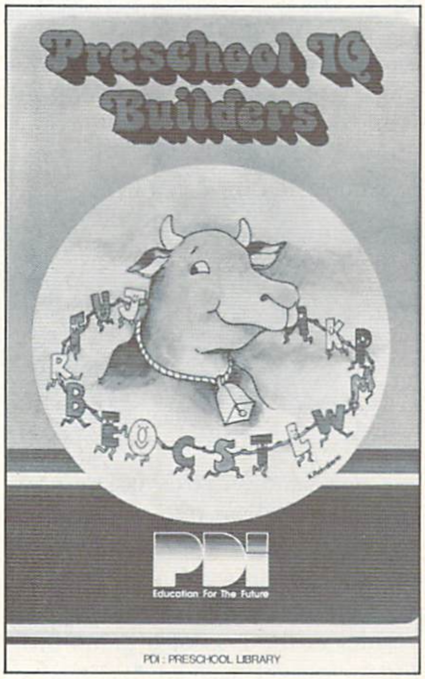
lichen
khaki
indigo
crimson

Chances are, your 11-year-old might have to mull that one over for a few seconds. It's just getting him to do that mulling that's the problem.

That's one problem, but there are others in this program as well. There is no music, no graphics, and limited use of color. Additionally, at times the words on the monitor were so blurry they were difficult to read. In all, there's little to convince the user that this is anything but work.

Work, as in salt mines, rock piles, garbage duty, and bedmaking. In fact, *Reading Comprehension: What's Different* reminds me a lot of my son's bedmaking ability: it's good, but it could be so much better

—Valerie B. Tamis



PRESCHOOL IQ BUILDER
PDI
C-64
Disk, cassette

Any preschooler who's grown up with *Sesame Street* will find the *PDI Pre-School IQ Builder I* program a piece of Big Bird's cake.

It's a fun learning experience that features happy faces and sad faces, with a musical background geared to the younger set. The object of this program is twofold. First, the child learns to recognize and differentiate between letters, numbers, and shapes. At the same time, his little fingers become familiar with all the letters on the keyboard of the computer. What more can you ask for in a preschool program?

In the first half of the program, "Same and Different," there are 6 separate lessons involving recognition and differentiation of letters, numbers, and shapes. The background music of *Here We Go 'Round the Mulberry Bush* sets the pace. The child must determine if the two characters on the monitor are similar, and if so press "s," or different, and press "l." If the child answers correctly *Mulberry Bush* continues and a happy face appears. If incorrectly, there's a sad face and the song stops. The computer keeps a tally of the number of right and wrong answers throughout the program. The child just has to push 0 to find out if he's made the Honor Roll or not. And believe me, they love it!

"Letter Builder," the second part of this program, is a bit more difficult. Here the child matches the letter on the screen with the same letter on the computer keyboard. Another oldie but goodie for the preschool set, *She'll Be Comin' 'Round the Mountain,* accompanies the happy and sad faces that appear on the screen.

In Lesson 1 the child learns "s" and "a," pressing the cor-

rect letter as it appears on the screen. (It is suggested in the accompanying pamphlet to this program that the parent cover all the other keys. I did not find that necessary to do.) In Lesson 2 "d" and "f" are added to "a" and "s." By Lesson 4 the child is hunting for and pecking "a,s,d,f,q,w,e,r,t,z,x,c,v,b," and by the fifth and final lesson all 26 letters of the alphabet are flashed on the monitor.

Now I know my ABC's? It's easy to find out because the computer keeps track of a child's correct and incorrect answers, and by pressing 0 he can immediately see his score.

Both "Same and Different" and "Letter Builder" are useful educational tools. They require concentration and patience as well as a parent's assistance and presence. Most likely a preschooler would not be able to load the programs himself, and he'd need and want you by his side as he pecks away and checks his score. So don't just hand him the disc and think you're going to settle down with the latest best seller.

However, as he masters the keyboard and moves on to more advanced programs (which he will do as a result of *this* program) that time will come!

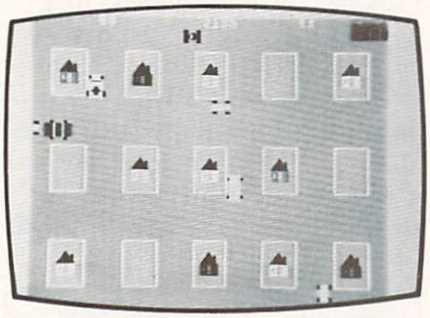
—Valerie B. Tamis

RESCUE SQUAD

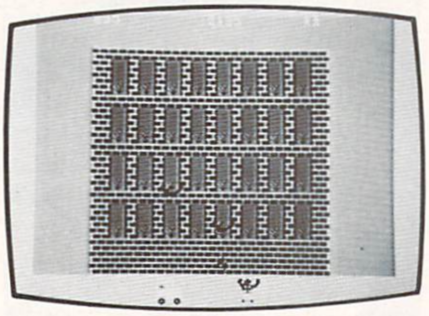
Muse Software
C-64
Disk; joystick

Your mission in *Rescue Squad* is threefold. You must safely maneuver your ambulance through rush-hour traffic to reach the burning building, you must catch the building's residents as they leap from windows to escape the flames, and, finally, you have to enter the building through a third floor window to rescue the remaining people trapped inside.

The problem with *Rescue Squad* is that each of the three screens or phases of your mission is like a mediocre Atari VCS game. When the game begins, your ambulance is in the lower left corner of the screen. The fire is in the upper right. In between are fifteen regularly spaced blocks that make up the simple grid of streets. What isn't so simple is avoiding the cars and trucks which zip back and forth along the narrow streets.



In **Rescue Squad,** *you must risk your neck in multiple scenarios.*



Unlike some mazes where you can actually lean into your turns before you actually reach the corner, here, any attempt to turn before you reach the cross street results in the loss of a game life. Once you get used to this, there's still the matter of the haphazard traffic to content with. Often, you'll be able to reach the fire with no problem. Other times, however, you'll lose most of your lives before even making it once. To a large extent, it depends on your luck, and this takes a lot of the fun out of the game.

The second screen requires you to catch people as they dive from windows. Although there is no penalty for letting people hit the pavement, you must avoid the flower pots which fall each time a person appears in one of the windows. Accidently catch three of the pots, and you move to the third screen.

Here, your job is to move your rescuer around a maze to reach and save the people who are still trapped. Also in the maze are deadly fireballs which move about randomly. When you reach one of the trapped residents, you have to carry him to safety before picking up another. To aid you, there are three asbestos jackets in the maze to temporarily protect you from the wandering flames. Although the number of fireballs increases with each level, the maze always remains the same. When all residents have been carried to safety, it's back to the first screen to begin again.

Rescue Squad isn't a bad game. In fact, you might even say that, on the whole, it's kind of fun. Unfortunately, we've seen it all before. The three screens taken separately are like *Dodge 'Em, Kaboom!,* and *Pac-Man,* only not nearly as good. If the game has a forte, it's the incredible music that accompanies your rescuing activities. It's reminiscent of the up-tempo piano from Keystone Cops chase scenes of the silent movie era. Great music, however, does not a great game make.

With all of the unique and entertaining games available for the C-64 now, it takes something really special to warrant spending

around $30 on a new game disk. *Rescue Squad* just doesn't have it.

—Lloyd Davies

DRAGONRIDERS OF PERN
Epyx
Commodore 64
Disc

After a long interval of four hundred years, the Red Star is close to Pern's orbit and threadfall is heavy. As the long silver spores of thread fall onto the holds of Pern, they scorch both flesh and land. As Weyrleader Benden, it is up to you to win the confidence of the major Lord Holders and Craftmasters of Pern through clever negotiating. Finally, you must mount your trusty dragon and lead the defense against the falling thread. If you're successful, you'll win the



Epyx Presient Michael Katz presents Anne McCaffrey, author of the* Dragonriders of Pern *series of adventure books, with the first copy of the computergame of the same name, fresh off the Epyx assembly line.

trust of Pern's leaders, and maybe even the game.

Dragonriders is a game for fantasy and adventure fans that like a little action now and then. It combines the strategic gameplay of D & D with an addicting action sequence in which you must maneuver a winged dragon through falling thread to destroy it with the dragon's breath.

During the first phase of the game, Negotiation/Intrigue, up to four players or "Weyrs" try to ally themselves with as many major holds and craft halls as possible. A Weyr is awarded two points for each major hold, and one point for every craft hall allied. To score diplomatic points and improve the chances of forming an alliance, each Weyr may negotiate individually with the holds, have weddings and invite influential guests they hope to impress, invite prospective allies to dragon hatchings, or make gifts of various objects that may be found as the game progresses.

The first negotiating screen is like a community calendar of upcoming events. It lists the six Weyrs and whatever negotiations, weddings, alliances, hatchings, etc. that they have scheduled, and on what day the events will occur. At the bottom right is the day clock which ticks away throughout the game. Beside the clock, messages are displayed to let everyone know the results of negotiations and attempted alliances, as well as any other news that may be of interest.

When a Weyr takes a turn, a second screen shows that player his status, lists his three strongest supporters, and displays the available options. The first option, Description, will give the Weyr a description of any Major Hold listed on the third screen. The descriptions are important because they give you clues to the proper negotiating strategy and attitude which is chosen from a fourth screen.

The Negotiation/Intrigue phase lasts for 240 days or about ten minutes in real time. This represents one year or turn in the game. Before the game begins, the total number of turns the game will last (from 1-99) is chosen. During the course of a game, however, each player may select one of the eight options whenever no other Weyr is currently in the selections process, or the player himself has no pending actions listed on the calendar of upcoming events. Also, if a Weyr is wounded in a duel (option #8), he will be ineligible for any negotiations until healed.

After this phase of the game, each Weyr's score is displayed and it's time to harness up the dragons and take to the sky. While threadfighting doesn't directly affect the score, Weyrs who skillfully defend against the thread will gain respect among the major holds when they return to the next Negotiation/Intrigue phase.

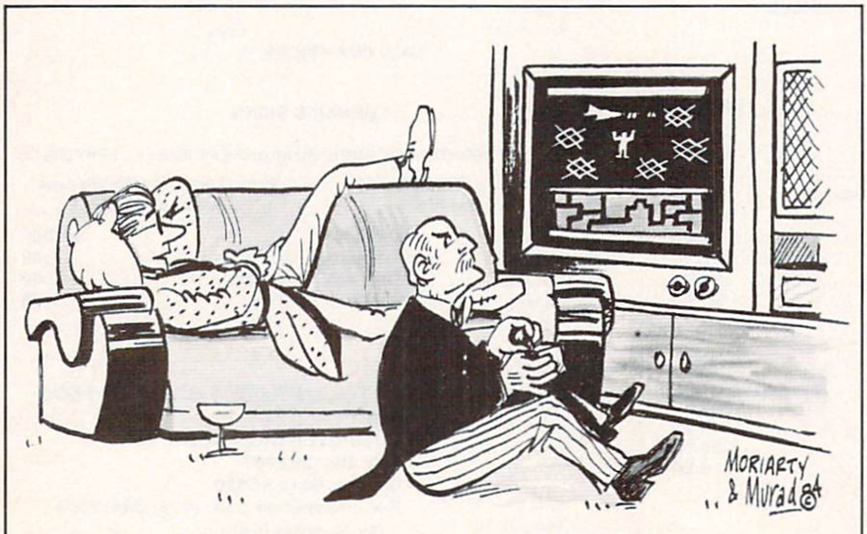
The game ends when a Weyr reaches 20 points, all of the preselected turns have been played out, or, because of sloppy threadfighting, too many holds have become infested.

This game can be frustrating at first, especially if you're not a big adventure fan. The directions can be confusing until you've gone through them a couple of times and played the game once or twice. Before long, though, you'll be wheeling and dealing your way around Pern, and battling thread like an old pro.

Fortunately, there are three varations of the game—the Standard Game (includes both Negotiation/Intrigue and Threadfighting), a game of Negotiation/Intrigue only with no Threadfighting, and Threadfighting Practice.

Dragonriders of Pern is a great adventure for anyone who'd like to explore the genre but isn't quite ready to give up the exciting graphics and acton of arcade type games. With *Dragonriders,* you get a nice taste of both worlds.

—Lloyd Davies



"Next time I'd like to get those bonus points, Jeeves!"

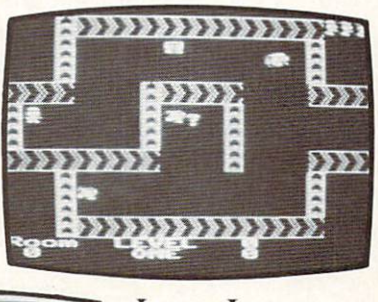
Continued from page 50
mania, with the creatures attacking your ground-based cannon every bit as bizarre and dangerous. For some reason, though, it seems old-fashioned now (my, how blase we've become in only a couple of years of computergaming!) and doesn't have the grabbing power of some other VIC games.

SECOND GENERATION

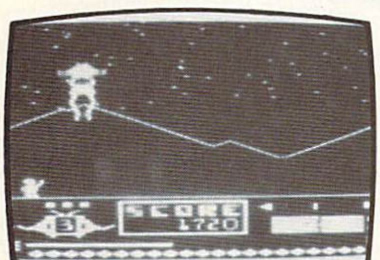
Going back to the Commodore label, there's a second generation of exciting game cartridges, and any reference to these must include the excellent rendition of the arcade favorite *Gorf.* I've seen versions of this game for other machines, and the VIC cartridge still has my vote as the best of the bunch. The four scenarios (Gorfs, Laser Attack, Space Warp, and Flag Ship) are difficult as the dickens to get through, but it's one of those games that you *have* to keep plugging away at no matter how late it is and no matter how early you have to get up the next morning. The graphics are excellent, and the game has all the elements of *Space Invaders, Star Raiders,* and a few others all rolled up into one neat game cartridge.

Another arcade game-turned VIC knockout is *Omega Race*—also in cartridge form from Commodore. Again, this is fairly faithful to the arcade game, and very exciting. The "rubber band" stretchiness of the center rectangle on the screen isn't quite the same as on the arcade game, but the concept is pretty faithful, and it's every bit as exciting.

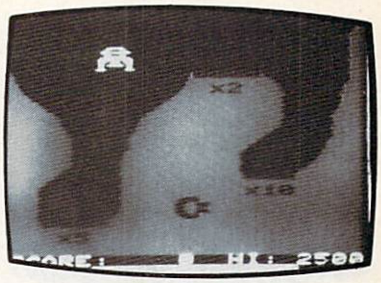
One unusual twist: because of the VIC's color capabilities, you can select from a wide variety of game backgrounds and frame color combinations—

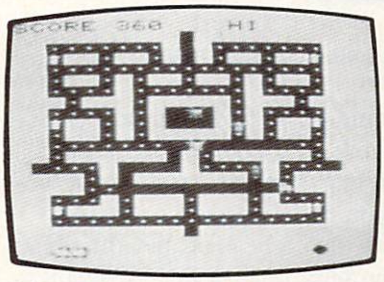


While it doesn't have even one hot tomato with legs that won't quit,* Shamus *feaures roomsful of deadly robodroids.



Lunar Leeper *sends your flying saucers skimming over the moonscape, rescuing humans and ducking lethal Leepers.*


Jupiter Lander *requires you to land your spacecraft safely, using retro-rockets when necessary to slow your descent.*



Trashman *is very much in the* Pac-Man *mode—except that you gobble your dots from behind the wheel of a garbage truck.*

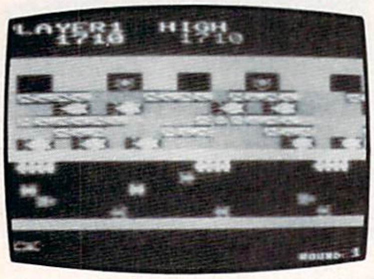
something you definitely can't do with the white-on-black screen of the arcade version. It's another winner, and you'll probably find some still around on dealers' shelves.

CHESS, ANYONE?

Commodore also offers a cartridge version of Hayden's *Sargon II Chess*—an outstanding computer chess game. Considering the fancy prices that game companies get for dedicated computerized chess games, getting this cartridge for the VIC will pay for the computer and then some. The program's chess-playing capability is excellent. If you play on higher skill levels, the computer can take its sweet time to make a move as it mulls over all the possible combinations. On the highest skill level, the VIC can actually take *several hours* to decide on a single move as it analyzes ahead by a dozen or more moves.

Since I never reached the expert class, I played *Sargon* at the lowest skill level, and felt that it was a pretty even match except for one thing—the computer *never* makes a mistake. While *Sargon's* play may be uninspired, it's always technically correct. You can beat the machine if you're good, very, very careful (don't dare make even one slip), and use just a touch of human inspiration to guide your game.

In the rehashed game circuit, there are some noteworthy VIC versions of games that have previously been available either on game-only machines or in other computer formats. Imagic's classic *Demon Attack* is excellent, although not much enhanced over its basic game machine version, which was excellent in itself. It's a super-grabber



Frogger ***in its VIC incarnation falls a few hops short of its Apple and Atari counterparts. But all the action is there!***

type of twitch game, and good for a few long nights.

Frogger has just come to us from Parker Brothers, and regrettably, it's not as good as the Apple and Atari computer versions. But all the action is there, and if you don't happen to have an Apple or Atari around, this is a good bet for you.

Another game in the new release pile is *Shamus* from HES (Human-Engineered Software). This is based on Synapse Software's very successful game for the Atari 400/800 and has the same general storyline, but is missing the depth of the graphics and convoluted mazes. It's a maze-type kill 'em before they kill you contest that can be engrossing.

NEW GAME SUPPLIER FOR THE VIC

One of the newer VIC cartridges is *Lunar Leeper* from Sierra On-Line. This game is best described as original, "cute," and hard as hell. Your saucer-shaped space ship skims over the lunar landscape where it encounters the Leepers, powerful, high-jumping creatures that eat spaceships for breakfast. There are various humans on the moon's surface waiting to be rescued by one of your saucers, if you can get to them without becoming a Leeper's ham and eggs and hashbrowns—a feat more easily said than done. Your spaceships have no brakes and are very hard to maneuver accurately.

One of the more original cartridges we've seen is Creative Software's *Rat Hotel,* in which your hero is a rat trying to eat the cheese deposited on each floor of the hotel while avoiding traps and the hotel security officer who won't hesitate to kill you. When you've eaten all the cheese, you ring for the elevator, and if your timing is right you can take the elevator to the next lower floor to eat some more. It's cute, but not earth-shaking; borderline reminiscent of Commodore's *Radar Rat Race,* an early VIC entry that features rats running a maze to find cheese while avoiding the usual enemies.

Much more challenging is the same Creative Software's *Trashman*—a transparent takeoff on *Pac-Man* and much more difficult to master. In this scenario, instead of a pie-shaped munchkin, you are a little garbage truck running around the maze collecting pellets of garbage. You have to keep ahead of the avenging rats which follow you through the maze with one purpose: to do you in.

There are power pellets in the corners (naturally) that let you turn the tables on your pursuers, and there are escape tunnels that bring you to the opposite side of the maze. If I hadn't already spent so much time playing the original, this game would be engrossing. If you've had little experience with *Pac-Man* (since you may not own an Atari), then *Trashman* will fill the bill nicely and help you work out your *Pac-Man* mania.

CASSETTES MAKE THE GRADE

There are also quite a few good games on cassette tape, but as you probably know, these take forever to load and frequently throw up loading error messages on the screen. Some of the better varieties of these cassettes require 8K expansion (we usually plug in a 16K module). That extra memory is really a must to make a game interesting and worthwhile. It stands to reason: complex instructions and detail need memory, and the VIC has only 3.5K of RAM. Personal favorites among these cassettes are *Sword of Fargoal* and *Ricochet* from Epyx.

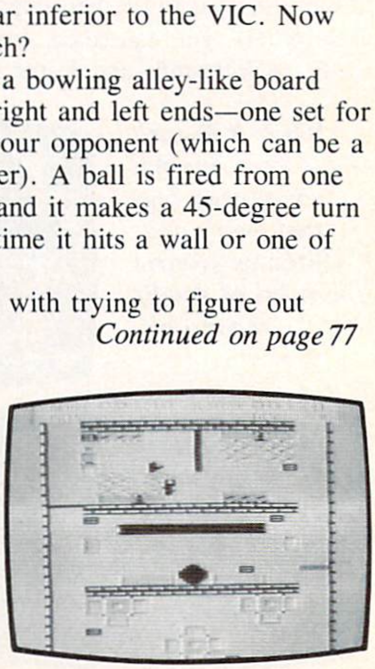
Ricochet is a fast-moving stratgegy board game. It's one of those games that is such a gem for the VIC that any copies pale by comparison. The software company later released a version for the Atari 400/800 which was far inferior to the VIC. Now how's that for a switch?

Ricochet gives you a bowling alley-like board with "pins" at both right and left ends—one set for you and one set for your opponent (which can be a human or the computer). A ball is fired from one corner at high speed and it makes a 45-degree turn or "ricochet" every time it hits a wall or one of the pins or bumpers.

The strategy comes with trying to figure out

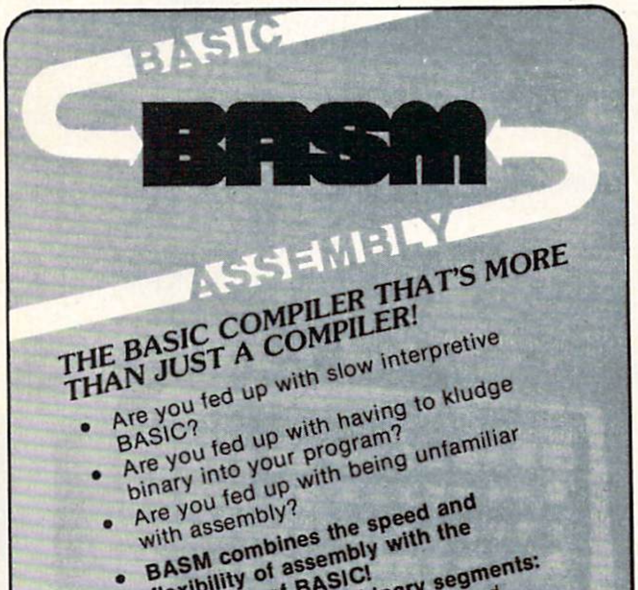
Continued on page 77

Creative Software's **Rat Hotel** ***is a people-infested hostelry that your furry persona may not live to check out of.***





"It's getting harder to tell the men from the boys by the price of their toys."



Reader Service No. 99

Stop Gambling. Start Winning.

You will beat the dealer if you play Blackjack correctly. They haven't changed the rules. You can win just as easily in 1984 as you could in 1961 when the first Blackjack strategies were created. Spend the time to learn how. It won't be easy, but will be well worth it.

BLACKJACK TEACHER simulates, in precise detail, all of the events that transpire in actual casino play. The display screen depicts the top view of the Blackjack table. You interact with the program just as you would an actual game. Computer controlled players participate to enhance the simulation. All events occur in real-time.

BLACKJACK TEACHER teaches seven different strategies, from Basic to Advanced Count. This spectrum of strategies allows you to select one that suits your needs.

BLACKJACK TEACHER monitors all of your betting and playing decisions to give you the feedback you need to learn effectively. By religiously practicing with the system, you will become an expert at the game faster than ever before possible.

BLACKJACK TEACHER is the result of over ten years of computer-assisted Blackjack research; it is by far the most comprehensive Blackjack instruction system available. And of course, complete documentation is included.

Note: a simplified version of the system which teaches only Basic Strategy is available for the VIC 20.

———FILL OUT AND MAIL TODAY———

Name ____________________
Address ____________________
City ____________________
State ____________ Zip ____________

☐ COMMODORE 64 ($59.95) ☐ Cassette
☐ VIC 20 ($19.95) ☐ Diskette

Send check or Money order payable to:

SOTA Enterprises, Inc.
833 Garfield Ave, Suite 101
South Pasadena, CA 91030

Postage Paid -
California residents add 6½% sales tax

Reader Service No. 100

ANATOMY

Continued from page 25

computer's buses and damage RAM or ROM chips. The second port is for a five pin audio/video connector to a monitor or stereo.

The third port is for serial (one bit at a time) input from and output to pheripheral devices. Up to five devices, including disk drives and a printer, can be daisy chained from this port. Since data is transmitted serially, only one of this port's pins (5) is actually used for data. The purpose of another pin (3) is to signal peripherals that data is about to come from the computer, and a third pin (4) acts as a timer for the data sent over the computer's serial bus. Next in line is the cassette port and last is the RS-232 port. The term RS-232 refers to an Electronic Industries Association standard for a certain type of interface for communications between electronic devices.

There you have the anatomy of a C-64. As a doctor can better better diagnose his patient's ailments because he is familiar with human anatomy, so will you and I now better understand the limits of and magic in our computers. Just as a physician strives to cure his patient without surgery, so can we use our skills to make our 64's perform useful tasks without going below their keyboards. □

RANDOM FILES

Continued from page 38

parts of your programs.

PROGRAM REVEALED

Line 100	Opens command channel to disk drive.
Line 110	Opens data channel.
Lines 120-130	Entering track and sector.
Line 135	If both are zeros, then end the program.
Line 140	Is it a valid track?
Lines 150-180	Is it a valid sector?
Line 210	Memory read from desired block.
Line 220	Beginning of loop to get all characters from buffer for desired block.
Lines 225-230	Decode into printable characters and print.
Lines 240-280	Print disk status and wait for key to be pressed to go back for new track, sector.
Line 300	Close channels—end of program.

Next month we will go into more detail about reading in portions of blocks. We will also include a block editor program, and will discuss its varied uses. Until then, keep those drives spinning! □

SEE PAGE 82

SPACE LANES

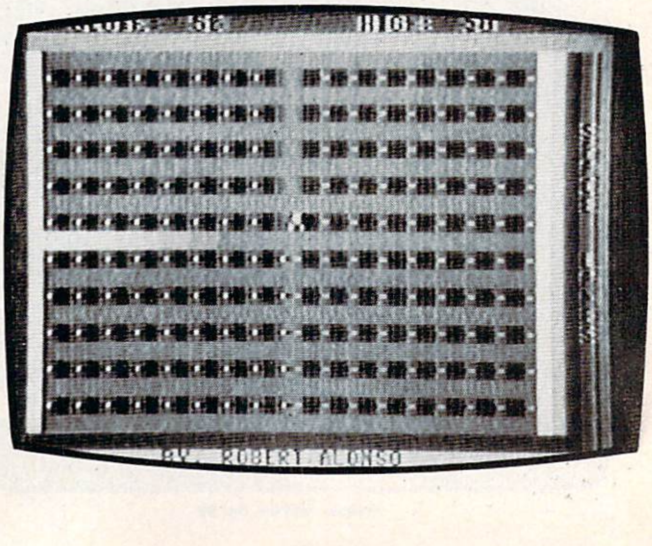
Continued from page 35

A verified copy will be sent within two days.

IMPORTANT! Hold the shift key down while typing any character underlined (C̲) in the program listing; hold the Commodore key down while typing any character overlined (C̄) in the listing.

PROGRAM REVEALED

Lines 1-5:	Initialization.
Lines 10-40:	Main game loop. All character movement takes place here.
Lines 50-65:	Laser subroutine.
Lines 90-96:	Scoring subroutine.
Lines 100-240:	Play screen setup. Maze is created and borders are defined.
Lines 500-580:	Title screen setup. First screen visible when program is run. In VIC-20 version the custom characters POKEing routine is included.
Lines 600-650:	Explosion sound subroutine.



Lines 660-680:	Closing routine. POKEs high score to memory location 900 and 901.
Lines 700-760:	(Used only in C-64 version.) Custom character routine and joystick matrix setup.
Line 800:	(C-64 only.) Data for joystick matrix.
Lines 810-820:	Data for spaceship and dot. □

SEE PAGE 79

VIC GAME BUYER'S GUIDE

Continued from page 74

where the ball will go with its many high-speed bounces, and moving your own paddle-shaped pins to try to deflect the ball in the best way. You use up a turn when you reset your paddles. It's interesting to watch the computer take a turn to reset also, and you can't help wondering if it's smarter than you are. It isn't; it just thinks faster. The computer keeps score flawlessly and fairly (we assume), and yes, you can win against the VIC. At the end of a match, it gives you a rating as a player, but the rating system it uses, like so many things in life, remains a mystery to us.

The *Sword of Fargoal* is another cassette that requires some memory expansion. It's an engrossing adventure-type maze game that lets you explore underground dungeons and fight off such adversaries as gargoyles, trolls, elvin rangers, mercenaries, werewolves, and werebears. You can pick up a lot of loot along the way, but can carry only 100 pieces of gold at a time unless you have a magic sack. You must bury the rest to retrieve later when you have "spent" some of your treasure as an offering in the temple.

You can occasionally refer to your map, and use a spell to help yourself out of a tight spot. By resting in the temple, you can regain some of your strength (after sacrificing some of your gold). But watch out for all those nasties! In addition to fighting you and sapping your strength, they'll try to steal your treasure.

This is just a once-over-lightly sampling of the VIC games this writer has really enjoyed. There are many, many more out there, and it's just a matter of finding them. You may discover some of them in low-priced sale bins. We've seen VIC game cartridges being liquidated at under $20 a pop in some retail outlets. For some reason, a lot of people no longer appreciate what a truly good game machine the VIC can be. But their loss is our gain. Go to it! □

THE CITY OF TRUTH

Continued from page 19

pond to the preceding paragraph. That should be very straightforward.

We are a long way from typing the puzzle mentioned at the beginning of this article into our computer and receiving an answer. However, we now have the groundwork to write programs for solving some types of logical problems. Good luck in finding your way to the City of Truth! □

PROGRAM LISTINGS

On the following pages, there are listed several programs that you might wish to try out on your own computer. Before you do so, there are a few things that you will need to know.

Certain computer commands are displayed on the monitor by various odd looking characters. To get your computer to *print* these command symbols (rather than perform the action of the command) you need to enter the quote mode. To do so, hold down the SHIFT key and press the "2" key; a pair of quotation marks will appear. This tells the computer that the next symbol is to be represented by a character. To get out of the quote mode just type in another set of quotation marks. You will also enter the quote mode when you INserT spaces or characters into a line. The easiest way to get out of the quote mode is to hit the RETURN key.

In Ahoy's program listings, you will run into letters and/or numbers surrounded by a pair of brackets. You will notice that these brackets appear neither on your keyboard nor in your printed programs. You, in all probability, use a dot matrix printer, but for the purpose of reproduction, we at ***AHOY!*** use a letter quality printer that is incapable of reproducing the command symbols.

Thus, when you are in the quote mode and press the SHIFT and CLR/HOME keys at the same time, your screen (and dot martix printer) would indicate this command with a character that looks like a heart ( ). Since a letter quality printer is unable to duplicate this symbol it substitutes an alternate code that is listed within the brackets{ } In the case of the SHIFT CLR/HOME symbol, our printer will print {SC}. What this command does is tell the computer to clear the screen, and return the cursor to the "home" position (top left of the screen).

An alternate way of entering the various commands listed below, as well as the several other graphic symbols and characters, is by typing in their appropriate character strings (CHR$). For example, the CLR/HOME command is CHR$(147). While typing character strings requires a few extra strokes, it does facilitate reading in the printed version, or when editing programs. For a complete list of CHR$ codes, consult the appendix in the back of your user manual

Below are listed a series of commands, the character that represents them on the screen or dot matrix printer, and how they appear in an ***AHOY!*** program listing.

When You See	It Means	You Type	You Will See	When You See	It Means	You Type	You Will See
{SC}	Screen Clear	SHIFT CLR/HOME		{YL}	Yellow	CNTRL 8	
{HM}	Home	CLR/HOME		{OR}	Orange	COMMODORE 1	
{CU}	Cursor Up	SHIFT ↑ CRSR ↓		{BR}	Brown	COMMODORE 2	
{CD}	Cursor Down	↑ CRSR ↓		{LR}	Light Red	COMMODORE 3	
{CL}	Cursor Left	SHIFT ← CRSR →		{G1}	Grey 1	COMMODORE 4	
{CR}	Cursor Right	← CRSR →		{G2}	Grey 2	COMMODORE 5	
{SS}	Shifted Space	SHIFT space		{LG}	Light Green	COMMODORE 6	
{IN}	Insert	INST		{LB}	Light Blue	COMMODORE 7	
{RV}	Reverse On	CNTRL 9		{G3}	Grey 3	COMMODORE 8	
{RO}	Reverse Off	CNTRL 0		{F1}	Function 1	F 1	
{BK}	Black	CNTRL 1		{F2}	Function 2	F 2	
{WH}	White	CNTRL 2		{F3}	Function 3	F 3	
{RD}	Red	CNTRL 3		{F4}	Function 4	F 4	
{CY}	Cyan	CNTRL 4		{F5}	Function 5	F 5	
{PU}	Purple	CNTRL 5		{F6}	Function 6	F 6	
{GN}	Green	CNTRL 6		{F7}	Function 7	F 7	
{BL}	Blue	CNTRL 7		{F8}	Function 8	F 8	

SPACE LANES

From page 77

VIC-20 VERSION

```
•1 POKE36879,56:GOSUB500
•2 PL=3:S=36878:S1=S-1:GOSUB100
•5 POKEL,.:POKEL+CC,7
•10 POKE37154,127:IFPEEK(37152)=11
•9THEND=1
•15 JR=PEEK(37137):IFJR=118ORJR=54
THEND=22
•20 IFJR=110ORJR=46THEND=-1
•25 IFJR=122ORJR=58THEND=-22
•30 L1=PEEK(L+D):IFL1=63THEND=0:GO
TO10
•35 IFL1=4THENPO=PO+5:GOSUB90
•38 IFINT(RND(0)*10)=9THENGOSUB50
•40 POKEL,32:L=L+D:POKEL,.:POKEL+C
C,7:D=0:GOTO10
•50 R=INT(RND(1)*10):R1=R*44
•55 FORX=7725+R1TO7741+R1:POKEX,63
:POKEX+CC,1:NEXTX
•60 IFL>7724+R1ANDL<7742+R1THENPL=
PL-1:GOSUB600
•65 FORX=7725+R1TO7741+R1:POKEX,32
:NEXTX:RETURN
•90 POKES,15:POKES,0:IFPO>HSTHENHS
=PO
•92 PRINT"{HM}{CR}{CR}{CR}{CR}{CR}
{CR}";PO:PRINT"{HM}":PRINTTAB(17)
"{CU}{CU}{CU}{CU}";HS
•94 IFPO/400=INT(PO/400)THENGOSUB4
00
•95 RETURN
•100 CC=30720:L=7931:D=0:HS=(PEEK(
901)*256)+PEEK(900):POKE36869,255
•110 PRINT"{SC}{CY}???????????????
???????"
•115 PRINT"{CU}{PU}???????????????
???????{CL}{CL}"
•120 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•130 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•135 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•140 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•145 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•150 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•155 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•160 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•165 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•170 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•175 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•180 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•185 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•190 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•195 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•200 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•205 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?"
•210 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ? {WH}?{GN}?{YL}?{PU}?"
•215 PRINT"{CU}{WH}?{CY}{17 spaces}
        {WH}?{GN}?{YL}?{PU}?{CL}{C
L}{CL}"
•220 PRINT"{CU}{GN}??????????????
?????{YL}?{PU}?{HM}"
•225 FORX=8164TO8185:POKEX,63:POKE
X+CC,3:NEXTX:POKE8184+CC,7:POKE81
85+CC,4
•238 PRINT"{HM}":PRINTTAB(21)"{BK}
{CD}{CD}{CD}S{CD}{CL}P{CD}{CL}A{C
D}{CL}C{CD}{CL}E{CD}{CD}{CL}M{CD}
{CL}A{CD}{CL}Z{CD}{CL}E"
•239 PRINT"{BK}{HM}{CD}{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{CD}{CR}{CR}BY, ROBERT ALONSO{
HM}"
•240 PRINT"{HM}{BK}SCORE:"::PRINT"
{HM}":PRINTTAB(12)"{CU}{CU}{CU}{C
U}{CU}HIGH:"
•400 FORX=7747TO7763STEP2:FORX1=0T
O352STEP44:POKEX+X1,4:POKEX+X1+CC
,7:NEXTX1,X
•420 RETURN
•500 PRINT"{SC}":FORX1=1TO200:PRIN
T"{HM}{CD}{CD}{CD}{CD}{CD}"
•505 DL=INT(RND(1)*500)+1:DC=INT(R
ND(1)*8)+1:POKE7680+DL,160:POKE38
```

```
•400+DL,DC
•507 PRINTTAB(8)
•510 FORX=0TO7:POKE646,X:PRINT"{CD
}{CL}{CL}{CL}{CL}{CL}{CL}{CL}{CL}
SPACE MAZE";:NEXT
•515 NEXTX1
•545 IFPEEK(7168)<>24THENPOKE900,0
:POKE901,0
•550 FORX=7168TO7679:POKEX,PEEK(X+
25600):NEXT
555 FORX=0TO7:READA:POKE7168+X,A:
NEXT
•557 FORX=0TO7:READA:POKE7200+X,A:
NEXT
•560 FORX=0TO7:POKE7672+X,255:NEXT
•570 RETURN
•600 POKES1,222:FORX=9TO15STEP2:PO
KES,X:FORX1=1TO45:NEXTX1,X:FORX2=
1TO50:NEXTX2
•605 FORX=15TO10STEP-1:POKES,X:FOR
X1=1TO50:NEXTX1,X:POKES,0
•610 FORX=15TO0STEP-1:POKES,X:FORX
1=1TO70:NEXTX1,X
•620 IFPL=0THEN660
•630 RETURN
•660 H1=INT(HS/256):H2=HS-H1*256
•670 POKE900,H2:POKE901,H1
•680 PRINT"{SC}":POKE36879,27:POKE
36869,240:END
•800 DATA24,60,126,24,189,231,195,
129
•810 DATA0,0,0,24,24,0,0,0
```

C-64 VERSION

```
•1 POKE54287,43:POKE54286,52:GOSUB
•500·GOTO700
•2 PL=3:GOSUB100
•5 POKEL,.:POKEL+CC,7
•10 J=15-(PEEK(56321)AND15)
•15 D=R(J)
•35 L1=PEEK(L+D):IFL1=63THEND=0:GO
TO10
•37 IFL1=4THENPO=PO+5:GOSUB90
•38 IFINT(RND(0)*10)=9THENGOSUB50
•40 POKEL,32:L=L+D:POKEL,.:POKEL+C
C,7:D=0:GOTO10
•50 R=3+2*INT(RND(0)*10)
•52 R1=((R-1)*40)
•55 FORX=1025+R1TO1058+R1:POKEX,63
:POKEX+CC,1:NEXTX
•60 IFL>1024+R1ANDL<1059+R1THENPL=
PL-1:GOSUB600
•65 FORX=1025+R1TO1058+R1:POKEX,32
:NEXTX:RETURN
•90 POKE54296,6:POKE54296,0:IFPO>H
STHENHS=PO
•92 PRINT"{HM}{YL}{CR}{CR}{CR}{CR}
{CR}{CR}{CR}{CR}{CR}{CR}";PO:POKE
1034,63:POKE1034+CC,7
•93 PRINTTAB(29)"{CU}{CU}{YL}";HS:
POKE1053,63:POKE1053+CC,7
•94 IFPO/900=INT(PO/900)THENGOSUB4
00
•96 RETURN
•100 CC=54272:L=1521:D=0:HS=(PEEK(
901)*256)+PEEK(900)
•102 DIMR(10):FORX=0TO10:READA:R(X
)=A:NEXTX
•105 POKE53265,PEEK(53265)AND239
•110 POKE53272,(PEEK(53272)AND240)
OR12
•120 PRINT"{CU}{YL}???????????????
??????????????????????????"
•130 PRINT"{CU}{PU}???????????????
??????????????????????????{CL}{CL}
"
•140 PRINT"{WH}? [34 spaces]
                    ?{CY}?{GN}?{YL}?{
PU}?"
•150 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•160 PRINT"{CU}{WH}? [34 spaces]
                      ?{CY}?{GN}?{Y
L}?{PU}?"
•170 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•180 PRINT"{CU}{WH}? [34 spaces]
                      ?{CY}?{GN}?{Y
L}?{PU}?"
•190 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•200 PRINT"{CU}{WH}? [34 spaces]
                      ?{CY}?{GN}?{Y
L}?{PU}?"
•210 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•220 PRINT"{CU}{WH}? [34 spaces]
                      ?{CY}?{GN}?{Y
L}?{PU}?"
•230 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? ? {WH}?{CY}
"?{GN}?{YL}?{PU}?"
```

```
•240 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•250 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•260 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•270 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•280 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•290 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•300 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•310 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•320 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•330 PRINT"{CU}{WH}? {RD}? ? ? ? ?
 ? ? ?  ? ? ? ? ? ? ? ? {WH}?{CY}
?{GN}?{YL}?{PU}?"
•335 PRINT"{CU}{WH}? {34 spaces}
?{CY}?{GN}?{Y
L}?{PU}?"
•350 PRINT"{CU}{GN}??????????????
??????????????????????{GN}?{YL}?{
PU}?{CL}{CL}{CL}"
•360 FORX=0TO39:POKE1984+X,63:POKE
1984+CC+X,3:NEXTX
•370 POKE2022+CC,7:POKE2023+CC,4
•380 PRINT"{HM}":PRINTTAB(38)"{YL}
{CD}{CD}{CD}{CD}S{CD}{CL}P{CD}{CL
}A{CD}{CL}C{CD}{CL}E{CD}{CD}{CD}{
CL}L{CD}{CL}A{CD}{CL}N{CD}{CL}E{C
D}{CL}S"
•390 PRINT"{HM}":PRINT"{CD}{CD}{CD
}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}
{CD}{CD}{CD}{CD}{CD}{CD}{CD}{CD}{
CD}{CD}{CD}{CY}{CR}{CR}{CR}{CR}{C
R}{CR}{CR}{CR}{CR}BY,?ROBERT?ALON
SO{HM}"
•395 PRINT"{HM}{YL}{CR}{CR}{CR}SCO
RE:":PRINT"{HM}"TAB(23)"{YL}HIGH:
"
•400 FORX=1145TO1161STEP2:FORX1=0T
O720STEP80:POKEX+X1,4:POKEX+X1+CC
,7:NEXTX1,X
•410 FORX=1162TO1178STEP2:FORX1=0T
O720STEP80:POKEX+X1,4:POKEX+X1+CC
,7:NEXTX1,X
•415 IFPO>0THENPO=PO+5
•420 POKE53265,PEEK(53265)OR16:RET
URN
•500 PRINT"{SC}{CD}{CD}{CD}{CD}{CD
}{CD}{CD}":POKE53280,8:POKE53281,
6
•520 PRINT"{WH}{RV}⊥  {CR}    {CR}⊥
* {CR}) * {CR}    {CR}{CR}{RD} {CR}
{CR}{CR}) * {CR} {CR} {CR}    {CR}⊥
  "
•530 PRINT"{CU}{WH}{RV} * {CR}{CR}
{CR} {CR}    {CR} {CR}{CR}{CR} {CR
}{CR}{CR}{CR}{RD} {CR}{CR}{CR}
{CR} * {CR} {CR}{CR}{CR} * "
•540 PRINT"{WH}*{RV} * {CR}    {CR}
{CR} {CR} {CR}{CR}{CR}  {CR}{CR}{
CR}{RD} {CR}{CR}{CR} {CR} {CR} {R
O}*{RV} {CR}   {CR}{CR}{RO}*{RV} *
"
•550 PRINT"{CU}{WH}{CR}*{RV} {CR}
{CR}{CR}{CR}   {CR} {CR}{CR}{CR}
{CR}{CR}{CR}{CR}{RD} {CR}{CR}{CR}
   {CR} {CR} {CR} {CR}{CR}{CR}{CR
}{RO}*{RV} "
•560 PRINT"{CU}{WH}{RV}  {RO}){RV}
{CR} {CR}{CR}{CR} {CR} {CR}{RO}*{
RV} {RO}){RV}{CR}   {CR}{CR}{RD}
  {CR} {CR} {CR} {CR} {CR}   {CR}
  {RO})"
•570 PRINTTAB(11)"{WH}{CD}{CD}BY,
ROBERT ALONSO"
•580 RETURN
•600 V=54296:FORX=54272TO54296:POK
EX,0:NEXTX
•605 POKE54273,30:POKE54272,25:POK
E54277,15:POKE54278,40
•610 POKE54276,129:FORX=9TO15STEP2
:POKEV,X:FORX1=1TO45:NEXTX1,X:POK
EV,0
•620 POKE54273,25:FORX=15TO10STEP-
1:POKEV,X:FORX1=1TO50:NEXTX1,X:PO
KEV,0:POKEV,1
•630 POKE54273,30:FORX=15TO0STEP-1
:POKEV,X:FORX1=1TO40:NEXTX1,X
•640 IFPL=0THEN660
•650 RETURN
•660 H1=INT(HS/256):H2=HS-H1*256
```

```
670 POKE900,H2:POKE901,H1
680 PRINT"{SC}":POKE53272,21:END
700 IFPEEK(12288)<>24THENPOKE900,
0:POKE901,0
704 FORX=0TO10:READA:NEXTX
705 POKE56334,PEEK(56334)AND254:P
OKE1,PEEK(1)AND251
710 FORX=12288TO12799:POKEX,PEEK(
X+41984):NEXTX
720 POKE1,PEEK(1)OR4:POKE56334,PE
EK(56334)OR1
730 FORX=0TO7:READA:POKE12288+X,A
:NEXTX
740 FORX=0TO7:READA:POKE12320+X,A
:NEXTX
750 FORX=0TO7:POKE12544+X,0:POKE1
2792+X,255:NEXTX
760 POKE52,48:POKE56,48:CLR:GOTO2
800 DATA0,-40,40,0,-1,-41,39,0,1,
-39,41
810 DATA24,60,126,24,189,231,195,
129
820 DATA0,0,0,24,24,0,0,0
```

RANDOM FILES

From page 76

```
100 OPEN15,8,15,"I"
110 OPEN5,8,5,"#"
120 PRINT"{SC}ENTER TRACK,SECTOR
(ENTER 0,0 TO END):"
130 INPUTT,S
135 IFT=0ANDS=0THEN300
140 IFT<1ORT>35THENPRINT"TRACK MU
ST BE FROM 1 TO 35":GOTO120
150 IFS<0ORS>20THENPRINT"SECTOR M
UST BE FROM 0 TO 20":GOTO120
160 IFT>30ANDS>16THENPRINT"TRACK"
T"CONTAINS SECTORS 0 TO 16":GOTO1
20
170 IFT>24ANDS>17THENPRINT"TRACK"
T"CONTAINS SECTORS 0 TO 17":GOTO1
20
180 IFT>17ANDS>18THENPRINT"TRACK"
T"CONTAINS SECTORS 0 TO 18":GOTO1
20
200 REM ** READ DATA FROM BLOCK *
*
210 PRINT#15,"U1:"5;0;T;S
220 FORX=0TO255:GET#5,A$:POKE212,
1
225 IFASC(A$+CHR$(0))<32THENPOKE2
12,0:PRINT"{RV}";:A$=CHR$(ASC(A$+
CHR$(0))+64)
230 PRINTA$;:POKE212,0:PRINT"{RO}
";:NEXT
240 POKE212,0:PRINT:PRINT"{CD}{CD
}DISK STATUS:"
250 INPUT#15,A,B$,C,D:PRINTA,B$,C
,D
260 PRINT"{CD}{RV}HIT A KEY TO CO
NTINUE"
270 GETA$:IFA$=""THEN270
280 GOTO120
300 CLOSE5:CLOSE15
```

DYNAMIC POWER

From page 48

LISTING 1

```
10 PRINTCHR$(147)"ENTER STARTING,
 ENDING LINE NUMBER:"
20 INPUTS,E
30 IFS>60000ORE>60000ORE<SORS<0OR
E<0THEN10
40 INPUT"INCREMENT LINES BY";IN
50 X=S
60 PRINTCHR$(147)X;"DATA";:I$=""
70 GETA$:IFA$=""THENPRINT"${CL}";
:GOTO70
75 IFLEN(I$)>65ANDA$<>CHR$(13)AND
A$<>CHR$(20)THEN70
80 IFA$=CHR$(20)ANDLEN(I$)=0THEN7
0
90 PRINTA$;
100 IFA$=CHR$(20)THENI$=LEFT$(I$,
LEN(I$)-1):GOTO70
110 IFA$=CHR$(13)THEN130
120 I$=I$+A$:GOTO70
130 PRINT"X=";X;":IN=";IN;":E=";E
:PRINT"GOTO150"
140 POKE631,19:POKE632,13:POKE633
,13:POKE634,145:POKE635,145:POKE6
36,13
145 POKE198,6:END
150 X=X+IN:IFX>ETHENEND
160 GOTO60
```

LISTING 2

```
60001 PRINTCHR$(147)"ENTER STARTI
NG, ENDING LINE NUMBER:":INPUTS,E
60002 IFS>60000ORE>60000ORE<SORS<
```

```
0ORE<0THEN60001
•60003 INPUT"INCREMENT LINES BY";I
N:X=S
•60004 PRINTCHR$(147)X;"DATA";:I$=
""
•60005 GETA$:IFA$=""THENPRINT"${CL
}";:GOTO60005
•60006 IFLEN(I$)>65ANDA$<>CHR$(13)
ANDA$<>CHR$(20)THEN60005
•60007 IFA$=CHR$(20)ANDLEN(I$)=0TH
EN60005
•60008 PRINTA$;:IFA$=CHR$(20)THENI
$=LEFT$(I$,LEN(I$)-1):GOTO60005
•60009 IFA$=CHR$(13)THEN60011
•60010 I$=I$+A$:GOTO60005
•60011 PRINT"X=";X;":IN=";IN;":E="
;E:PRINT"GOTO60013":POKE631,19:PO
KE632,13
•60012 POKE633,13:POKE634,145:POKE
•635,145:POKE636,13:POKE198,6:END
•60013 X=X+IN:IFX>ETHENEND
•60014 GOTO60004
```

PRIME NUMBER GENERATOR

From page 63

```
•1 REM PRIME NUMBER GENERATOR
•2 REM  BASED ON KMMM PASCAL DEMO
•3 REM WILSERV INDUSTRIES
•4 REM BELLMAWR N.J.
•5 REM MODIFIED BY MORTON A. KEVEL
SON
•6 REM FOR DTL-BASIC 64 BENCHMARK
TESTS
•7 REM CONVERT EXCLUDING INTEGER D
IRECTIVE
•8 REM ** CE ()
•9 T1=TI:REM JIFFY CLOCK VARIABLE
•10 L=1000:H=1
•20 DIM S(L)
•30 FOR J=1 TO L:S(J)=1:NEXT J
•35 PRINT "ARRAY CLEARED"
•40 FOR N=2 TO L
•50 IF S(N)<H THEN 80
•60 PRINT N;
•70 FOR J=N TO L STEP N:S(J)=0:NEX
T J
•80 NEXTN
•85 TE=TI-T1:PRINT:PRINT TE;"JIFFI
ES"
•87 REM 1 JIFFY = 1/60 SECOND
•90 END
```

ADDRESS BOOK

From page 40

```
•1500 DATA1500,DO,NOT,ENTER,DATA,O
N,THIS,LINE
•1501 PRINT"{SC}{WH}":POKE53281,0:
POKE53280,11
•1505 INPUT"{CD}{CD}{CD}{CD}{CD}{C
D}WHAT IS YOUR FIRST NAME";N$:GOT
01515
•1510 INPUT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}SORRY, I FORGOT YOUR NAME";
N$
•1515 PRINT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}"SPC(20-INT((LEN(N$)+11)/2)
);"THANK YOU ";N$;"."
•1520 FORL=1TO1000:NEXT
•1525 PRINT"{SC}{WH}":POKE53281,0:
POKE53280,11
•1526 PRINT"{YL}ADDRESS BOOK <C> 1
984 PLANET EARTH LTD.{WH}":PRINT:
T=0
•1530 PRINT"[L]  LIST ALL ENTRIES"
•1531 PRINT
•1532 PRINT"[N]  LAST NAME"
•1533 PRINTSPC(25);"{RV}
{RO}"
•1534 PRINT"[F]  FIRST NAME"SPC(10
);"{RV} ADDRESS  {RO}"
•1535 PRINTSPC(25);"{RV}      BOOK
{RO}"
•1536 PRINT"[A]  ADDRESS"SPC(12);"
O{RV}  W    W  {RO}P"
•1537 PRINTSPC(23);"{RV}{G2} {WH}{
RO}%  {RV}{G2} {RO}     {RV} {RO}{
WH}  '{RV}{G2} {RO}{WH}"
•1538 PRINT"[C]  CITY"SPC(15);"L{R
V}  W    W  {RO}:"
•1539 PRINTSPC(24);"{RV}{BR} {WH}B
Y          {BR} {RO}{WH}"
•1540 PRINT"[S]  STATE"SPC(14);"{R
V}{BR} {WH} B.BEHLING{BR} {RO}{WH
}"
•1541 PRINTSPC(24);"{RV}{BR} {WH}C
CCCCCCCCCCC{BR} {RO}{WH}"
•1542 PRINT"[Z]  ZIP CODE"SPC(11);
"{RV}{BR}             {RO}{WH}"
```

```
•1543 PRINTSPC(24);"{RV}{BR}
      {RO}{WH}"
•1544 PRINT"[P]  PHONE NUMBER"SPC(
7);" {RV}{G2}I{RO}         {RV}I{R
O}{WH}"
•1545 PRINT
•1546 PRINT"[X]  EXIT PROGRAM"
•1547 PRINT
•1548 PRINT"[E]  ENTER DATA"
•1549 PRINT
•1550 PRINT"{LG}CHOICE?{WH}"
•1592 GETK$:IFK$=""THEN1592
•1605 IFK$="L"THENGOSUB1722
•1610 IFK$="N"THENGOSUB2000
•1615 IFK$="F"THENGOSUB7000
•1620 IFK$="A"THENGOSUB9000
•1625 IFK$="C"THENGOSUB4000
•1630 IFK$="S"THENGOSUB10000
•1635 IFK$="X"THENGOSUB6000
•1640 IFK$="E"THENGOTO23000
•1645 IFK$="Z"THENGOSUB12000
•1650 IFK$="P"THENGOSUB15000
•1721 GOTO1525
•1722 PRINT"{SC}"
•1723 GOSUB20000
•1724 IFA=1500THENGOTO1730
•1726 GOSUB19000
•1727 GOTO1722
•1730 PRINT"{SC}"SPC(12);"{CD}{CD}
{CD}{CD}{CD}{CD}{RV}{YL} END OF F
ILES. {RO}{WH}"
•1735 FORL=1TO1000:NEXT
•1740 RESTORE
•1745 RETURN
•2000 INPUT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}WHAT IS THE LAST NAME";J$:Z
=LEN(J$)
•2010 PRINT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}"SPC(20-INT((LEN(N$)+11)/2)
);"THANK YOU ";N$;"."
•2020 FORL=1TO1000:NEXT:GOTO2060
•2030 PRINTSPC(7);"{CD}{CD}{CD}{CD
}SEARCHING, PLEASE STANDBY."
•2060 GOSUB20000
•2070 IFA=1500THENGOTO2140
•2090 IFJ$=LEFT$(B$,Z)THENT=1:GOSU
B19000
•2130 PRINT"{SC}":GOTO2030
•2140 IFT=1THENGOTO2220
•2150 PRINT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}"SPC(20-INT((LEN(J$)+14+LEN
(N$))/2));J$;" NOT ON FILE "N$;".
"
•2160 PRINTSPC(8);"{CD}{CD}{YL}{RV
}£                            *"
•2170 PRINTSPC(8);"{RV} {RO}{WH}HI
T RETURN TO CONTINUE.{RV}{YL} "
•2180 PRINTSPC(8);"{RO}*{RV}
                   {RO}\{WH}"
•2190 GETK$:IFK$=""THENGOTO2190
•2200 RESTORE
•2210 RETURN
•2220 PRINT"{SC}"SPC(12);"{CD}{CD}
{CD}{CD}{CD}{CD}{RV}{YL} END OF F
ILES. {RO}{WH}"
•2230 FORL=1TO1000:NEXT
•2240 RESTORE
•2250 RETURN
•4000 INPUT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}WHAT CITY DO YOU WANT";J$:Z
=LEN(J$)
•4200 PRINT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}"SPC(20-INT((LEN(N$)+11)/2)
);"THANK YOU ";N$;"."
•4300 FORL=1TO1000:NEXT:GOTO4500
•4400 PRINTSPC(7);"{CD}{CD}{CD}{CD
}SEARCHING PLEASE STANDBY."
•4500 GOSUB20000
•4600 IFA=1500THENGOTO5300
•5000 IFJ$=LEFT$(E$,Z)THENT=1:GOSU
B19000
•5200 PRINT"{SC}":GOTO4400
•5300 IFT=1THENGOTO5750
•5350 PRINT"{SC}{CD}{CD}{CD}{CD}{C
D}{CD}"SPC(20-INT((LEN(J$)+14+LEN
(N$))/2));J$;" NOT ON FILE "N$;".
"
•5360 PRINTSPC(8);"{CD}{CD}{RD}{RV
}£                            *"
•5400 PRINTSPC(8);"{RV} {RO}{WH}HI
T RETURN TO CONTINUE.{RV}{RD} "
•5410 PRINTSPC(8);"{RO}*{RV}
                   {RO}£{WH}"
•5500 GETK$:IFK$=""THENGOTO5500
•5600 RESTORE
•5700 RETURN
•5750 PRINT"{SC}"SPC(12);"{CD}{CD}
{CD}{CD}{CD}{CD}{RV}{YL} END OF F
ILES. {RO}{WH}"
•5800 FORL=1TO1000:NEXT
•5850 RESTORE
•5900 RETURN
•6000 PRINT"{SC}{CD}{CD}{CD}{CD}"S
PC(5);"HIT [X] TO EXIT WITHOUT SA
VING."
•6100 PRINT"{CD}{CD}"SPC(8);"HIT [
```

```
A] TO ABORT TO MENU."
6200 PRINT"{CD}{CD}"SPC(8);"HIT [S] TO SAVE LISTINGS."
6350 PRINT"{CD}{CD}"SPC(5);"{RV}{YL} REWIND TAPE BEFORE SAVING!!!{RO}{WH}"
6900 GETK$:IFK$=""THENGOTO6900
6910 IFK$="X"THENEND
6920 IFK$="A"GOTO1525
6950 IFK$="S"THENSAVE"ADDRESS BOOK 4:1"
6960 RETURN
7000 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}WHAT IS THE FIRST NAME";J$:Z=LEN(J$)
7100 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(20-INT((LEN(N$)+11)/2));"THANK YOU ";N$;"."
7200 FORL=1TO1000:NEXT:GOTO7500
7300 PRINTSPC(7);"{CD}{CD}{CD}{CD}SEARCHING, PLEASE STANDBY."
7500 GOSUB20000
7600 IFA=1500THENGOTO8300
8000 IFJ$=LEFT$(C$,Z)THENT=1:GOSUB19000
8200 PRINT"{SC}":GOTO7300
8300 IFT=1THENGOTO8700
8350 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(20-INT((LEN(J$)+14+LEN(N$))/2));J$;" NOT ON FILE "N$;"."
8360 PRINTSPC(8);"{CD}{CD}{CY}{RV}£                  *"
8400 PRINTSPC(8);"{RV} {RO}{WH}HIT RETURN TO CONTINUE.{RV}{CY} "
8410 PRINTSPC(8);"{RO}*{RV}                  {RO}£{WH}"
8500 GETK$:IFK$=""THENGOTO8500
8600 RESTORE
8650 RETURN
8700 PRINT"{SC}"SPC(12);"{CD}{CD}{CD}{CD}{CD}{CD}{RV}{YL} END OF FILES. {RO}{WH}"
8750 FORL=1TO1000:NEXT
8800 RESTORE
8850 RETURN
9000 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}WHAT IS THE ADDRESS";J$:Z=LEN(J$)
9010 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(20-INT((LEN(N$)+11)/2));"THANK YOU ";N$;"."
9015 FORL=1TO1000:NEXT:GOTO9035
9020 PRINTSPC(7);"{CD}{CD}{CD}{CD}SEARCHING, PLEASE STANDBY"
9035 GOSUB20000
9040 IFA=1500THENGOTO9065
9045 IFJ$=LEFT$(D$,Z)THENT=1:GOSUB19000
9060 PRINT"{SC}":GOTO9035
9065 IFT=1THENGOTO9100
9070 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(1);J$;" NOT ON FILE ";N$;"."
9073 PRINTSPC(8);"{CD}{CD}{PU}{RV}£                  *"
9075 PRINTSPC(8);"{RV} {RO}{WH}HIT RETURN TO CONTINUE.{RV}{PU} "
9077 PRINTSPC(8);"{RO}*{RV}                  {RO}£{WH}"
9080 GETK$:IFK$=""THENGOTO9080
9085 RESTORE
9090 RETURN
9100 PRINT"{SC}"SPC(12);"{CD}{CD}{CD}{CD}{CD}{CD}{RV}{YL} END OF FILES. {RO}{WH}"
9110 FORL=1TO1000:NEXT
9120 RESTORE
9130 RETURN
10000 INPUT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}WHAT STATE DO YOU WANT";J$:Z=LEN(J$)
10010 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(20-INT((LEN(N$)+11)/2));"THANK YOU ";N$;"."
10015 FORL=1TO1000:NEXT:GOTO10035
10020 PRINTSPC(7);"{CD}{CD}{CD}{CD}SEARCHING, PLEASE STANDBY."
10035 GOSUB20000
10040 IFA=1500THENGOTO10065
10045 IFJ$=LEFT$(F$,Z)THENT=1:GOSUB19000
10060 PRINT"{SC}":GOTO10035
10065 IFT=1THENGOTO10095
10070 PRINT"{SC}{CD}{CD}{CD}{CD}{CD}{CD}"SPC(20-INT((LEN(J$)+14+LEN(N$))/2));J$;" NOT ON FILE "N$;"."
10073 PRINTSPC(8);"{CD}{CD}{GN}{RV}£                  *"
10075 PRINTSPC(8);"{RV} {RO}{WH}HIT RETURN TO CONTINUE.{RV}{GN} "
10077 PRINTSPC(8);"{RO}*{RV}                  {RO}£{WH}"
10080 GETK$:IFK$=""THENGOTO10080
10085 RESTORE
```

```
•10090 RETURN
•10095 PRINT"{SC}"SPC(12);"{CD}{CD
}{CD}{CD}{CD}{CD}{RV}{YL} END OF
FILES. {RO}{WH}
•10100 FORL=1TO1000:NEXT
•10105 RESTORE
•10110 RETURN
•12000 INPUT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}WHAT IS THE ZIP CODE";J$:Z
=LEN(J$)
•12005 PRINT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}"SPC(20-INT((LEN(N$)+11)/2
));"THANK YOU ";N$;"."
•12010 FORL=1TO1000:NEXT:GOTO12015
•12013 PRINTSPC(7);"{CD}{CD}{CD}{C
D}SEARCHING, PLEASE STANDBY."
•12015 GOSUB20000
•12020 IFA=1500THENGOTO12045
•12025 IFJ$=LEFT$(G$,Z)THENT=1:GOS
UB19000
•12040 PRINT"{SC}":GOTO12015
•12045 IFT=1THENGOTO12075
•12050 PRINT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}"SPC(20-INT((LEN(J$)+14+LE
N(N$))/2));J$;" NOT ON FILE "N$;"
."
•12053 PRINTSPC(8);"{CD}{CD}_{BL}{R
V}£                      *"
•12055 PRINTSPC(8);"{RV} {RO}{WH}H
IT RETURN TO CONTINUE.{RV}{BL} "
•12057 PRINTSPC(8);"{RO}*{RV}
                     {RO}){WH}"
•12060 GETK$:IFK$=""THENGOTO12060
•12065 RESTORE
•12070 RETURN
•12075 PRINT"{SC}"SPC(12);"{CD}{CD
}{CD}{CD}{CD}{CD}{RV}{YL} END OF
FILES. {RO}{WH}"
•12080 FORL=1TO1000:NEXT
•12085 RESTORE
•12090 RETURN
•15000 INPUT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}WHAT PHONE NUMBER";J$:Z=LE
N(J$)
•15005 PRINT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}"SPC(20-INT((LEN(N$)+11)/2
));"THANK YOU ";N$;"."
•15010 FORL=1TO1000:NEXT:GOTO15015
•15013 PRINTSPC(7);"{CD}{CD}{CD}{C
D}SEARCHING, PLEASE STANDBY."
•15015 GOSUB20000
•15020 IFA=1500THENGOTO15045
•15025 IFJ$=LEFT$(H$,Z)THENT=1:GOS
UB19000
•15040 PRINT"{SC}":GOTO15015
•15045 IFT=1THENGOTO15075
•15050 PRINT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}"SPC(20-INT((LEN(J$)+14+LE
N(N$))/2));J$;" NOT ON FILE "N$;"
."
•15053 PRINTSPC(8);"{CD}{CD}_{OR}{R
V}£                      *"
•15055 PRINTSPC(8);"{RV} {RO}{WH}H
IT RETURN TO CONTINUE.{RV}{OR} "
•15057 PRINTSPC(8);"{RO}*{RV}
                     {RO}£{WH}"
•15060 GETK$:IFK$=""THENGOTO15060
•15065 RESTORE
•15070 RETURN
•15075 PRINT"{SC}"SPC(12);"{CD}{CD
}{CD}{CD}{CD}{CD}{RV}{YL} END OF
FILES. {RO}{WH}"
•15080 FORL=1TO1000:NEXT
•15085 RESTORE
•15090 RETURN
•19000 PRINT"{SC}{CD}{CD}{CR}{LB}F
ILE NUMBER  >{WH}";A
•19010 PRINT"{CY}UCCCCCCCCCCCCI{WH
}"
•19100 PRINT"{CY}B{WH}NAME
{CY}B>{WH}";B$;", ";C$
•19110 PRINT"{CY}B                B{WH
}"
•19150 PRINT"{CY}B{WH}ADDRESS
{CY}B>{WH}";D$
•19160 PRINT"{CY}B                B{WH
}"
•19200 PRINT"{CY}B{WH}CITY
{CY}B>{WH}";E$
•19210 PRINT"{CY}B                B{WH
}"
•19300 PRINT"{CY}B{WH}STATE
{CY}B>{WH}";F$
•19310 PRINT"{CY}B                B{WH
}"
•19350 PRINT"{CY}B{WH}ZIP CODE
{CY}B>{WH}";G$
•19355 PRINT"{CY}B                B{WH
}"
•19356 PRINT"{CY}B{WH}PHONE NUMBER
{CY}B>{WH}";H$
•19357 PRINT"{CY}B                B{WH
}"
•19400 PRINT"{YL}QCCCCCCCCCCCCCCCC
CCCCCCCCI{WH}"
•19410 PRINT"{YL}B{WH}HIT [X] TO A
```

```
BORT TO MENU{YL}B"
•19420 PRINT"{YL}B
         B"
•19425 PRINT"B{WH}HIT [E] TO EDIT
FILE{YL}     B"
•19430 PRINT"B
     B"
•19440 PRINT"B{WH}HIT RETURN TO CO
NTINUE  {YL}B"
•19450 PRINT"JCCCCCCCCCCCCCCCCCCCC
CCCCK{WH}"
•19500 GETK$:IFK$=""THENGOTO19500
•19540 IFK$="X"THENRESTORE
•19550 IFK$="X"THENGOTO1525
•19560 IFK$="E"THENED=1:GOTO19700
•19600 RETURN
•19700 GOSUB23070
•19705 IFED=2THENPRINT"{CD}LIST";A
;"-";A+1:GOTO19715
•19710 PRINT"{CD}LIST"A
•19715 PRINT"{CU}{CU}{CU}";:POKE19
8,4:FORI=631TO636:POKEI,13:NEXT:E
ND
•20000 READA
•20100 IFA=1500THENRESTORE
•20110 IFA=1500THENRETURN
•20115 X=A
•20200 READB$,C$,D$,E$,F$,G$,H$
•20300 RETURN
•23000 INPUT"{SC}{CD}{CD}LAST NAME
";I1$
•23010 INPUT"{SC}{CD}{CD}FIRST NAM
E";I2$
•23015 INPUT"{SC}{CD}{CD}ADDRESS";
I3$
•23020 INPUT"{SC}{CD}{CD}CITY";I4$
•23025 INPUT"{SC}{CD}{CD}STATE";I5
$
•23030 INPUT"{SC}{CD}{CD}ZIP CODE"
;I6$
•23035 INPUT"{SC}{CD}{CD}PHONE NUM
BER";I7$
•23045 PRINT"{SC}{CD}{CD}{CD}";SPC
(10);"FILING. PLEASE WAIT."
•23050 GOSUB20000
•23055 IFX>=1500THENGOTO23120
•23060 IFA=1500THENGOTO23070
•23065 GOTO23050
•23070 LD=LEN(B$)+LEN(C$)+LEN(D$)+
LEN(E$)+LEN(F$)+LEN(G$)+LEN(H$)+L
EN(STR$(A))
•23075 IFLD>=67THENX=X+1:IFED=1THE
NED=2:RETURN
•23076 IFED=1THENRETURN
•23080 DL=LEN(I1$)+LEN(I2$)+LEN(I3
$)+LEN(I4$)+LEN(I5$)+LEN(I6$)+LEN
(I7$)
•23085 LI=X+1:DL=DL+LEN(STR$(LI))
•23090 R=4:IFDL>=67THEN23130
•23095 PRINT"{SC}{CD}{CD}{CD}"RIGH
T$(STR$(LI),LEN(STR$(LI))-1)"DATA
";LI;"{CL},";I1$;",";I2$;
•23100 PRINT",";I3$;",";I4$;",";I5
$;",";I6$;",";I7$
•23105 PRINT"{CD}RUN1510"
•23110 PRINT"{HM}";:POKE198,R:FORI
=631TO635:POKEI,13:NEXT:END
•23120 PRINT"{SC}{CD}{CD}{CD}{CD}{
CD}{CD}"SPC(13);"{RV}{LG}MEMORY F
ULL!!!{RO}{WH}":FORL=1TO2000:NEXT
•23125 RESTORE:RETURN
•23130 PRINT"{SC}{CD}{CD}{CD}"RIGH
T$(STR$(LI),LEN(STR$(LI))-1)"DATA
";LI;"{CL},";I1$;",";I2$;
•23135 PRINT",";I3$
•23140 LI=LI+1
•23145 PRINTRIGHT$(STR$(LI),LEN(ST
R$(LI))-1)"DATA";I4$;",";I5$;",";
I6$;",";I7$
•23150 R=5:GOTO23105
```

Access. Retrieval of information from computer memory.

Access time. Time needed to retrieve information from computer memory.

Address. Location (identified by a number) of a byte of information in the computer's memory.

Alphanumeric. Alphabetic and numeric characters.

Analog signal. A continuous electronic signal of any frequency or strength that represents a condition (i.e.: the position of the game control paddles).

Applications software. Programs that will instruct the computer to perform either a specific task or a series of related tasks, usually relating to business or home uses.

ASCII. The American Standard Code for Information Interchange. This code allows two computers, which may entertain different languages, to communicate.

Assembler. A program that converts information in the computer's memory into a binary code for proper execution.

Assembly language. Machine language that has been translated into mnemonic codes, thus making it easier for the programmers to remember. A three letter code would be the equivalent to a string of eight digits.

BASIC. Beginner's All-purpose Symbolic Instruction Code. Easy to use, popular programming language that is widely used with microcomputers.

Baud. A unit (bits per second) that measures the rate of speed at which data is translated from one device to another. (Heck, every baud *we've* ever known has been fast.)

Binary code. A system of numbers that uses two digits, 0 and 1, to express all characters (both numbers and letters). The computer then uses this string of numbers to process information.

Bit. An acronym for *B*inary Dig*IT*. Represents either "0" or "1" in the binary code. Approximately eight bits (one byte) are required to represent one character.

Bidirectional printing. Special feature on some computer printers that allows the printer to print first right to left and then drop to the next line and print left to right. This feature speeds up the printing process.

Board. A thin, rectangular, flat electronic component that contains one or more layers of printed circuits. Chips and other electronic parts are often attached to a circuit board.

Boot. To start or restart a computer by transferring instructions ("Heel!"?) from a storage device, such as a tape or disk, into the computer's memory.

Bug. A logic error in a computer program that causes the program to dysfunction.

Bus. A conductor that allows data to be passed between the various computer components. Buses are manufactured that will allow different brands of components to be used with the same computer. (They can often be seen entering terminals.)

Byte. Plural of bit. There are eight bits in one byte. Each byte represents one character.

CAI. Computer Assisted Instruction. This refers to a variety of instructional and/or educational software.

Card. A circuit board which is attached to a standing computer, often to boost its memory capability.

Cartridge. Sometimes referred to as a ROM module or solid state cartridge. Device that contains a prerecorded program (such as a game).

Cassette Tape Recorder. Device used either to store data or to house prerecorded programs. Often, but not always, the same type used in audio recording. If so, a special interface may still be required. A much slower and less reliable medium than disk drive.

C-BASIC. Very popular non-interactive language that is utilized by 8080, 8085 and Z80 microprocessor computers. C-BASIC executes data at a more accelerated rate than the standard interpreter BASIC.

Character. A single symbol, letter, or number.

Character set. Total catalogue of alphanumeric, special, and punctuation characters accessible to a printer or terminal.

Chip. An integrated circuit, a quarter to three quarters of an inch square, that is etched on a tiny piece of silicon. Chips are the building blocks of computers and are able to contain anywhere from a few dozen to several thousand transistors or circuit elements. They perform many functions: calculations, memory operation and storage or controlling other chips. Up to 32,000 bytes of information may be stored in a single chip.

CPU. Central Processing Unit, or microprocessor. The electronic "core" of a computer. All information is passed through the CPU and all functions are executed from the CPU.

Circuit Board. see board.

COBOL. Acronym for Common Business Oriented Language. High level language for business applications.

Command. An instruction (e.g.: run, load) that tells the computer to do something.

Compatibility. The ability of the computer and any or all of its peripherals to be able to function in conjunction with one another.

Compiler. A program that translates any high level language into the binary code that is required for the computer.

Computer. A programmable electronic device, consisting of a CPU, memory and input/output (I/O) capability, that stores, retrieves, and processes data.

Computer system. A computer setup that consists of a computer terminal and its software and various peripherals.

CP/M. Control program for microcomputers. A single-user operating system for microcomputers that is in very wide use, especially for business applications.

CRT. Cathode Ray Tube. The most popular type of display screen. Often it is simply the screen of an ordinary television set though it can be a monitor specially designed for use with computers.

Cursor. Symbol, most often a small flashing square, that indicates where the next character will appear.

Data. All information that is processed by the computer.

Datum. Singular of data.

Data Bank. The central storage locations for all of the information that the computer has access to.

Data base manager. Program that enables user to enter files, organize matter and sort and retrieve information.

Daisy wheel printer. A letter quality impact printer that utilizes a daisy wheel, a circular printing element containing as many as a hundred characters.

Descenders. The lower case characters g,j,p,q, and y, which have tails that extend past the baseline formed by the other characters.

Disk. A flat, round device for storing computer data. Like a phonograph disc, it rotates and contains information in tracks. Unlike a phonograph disc, it is never removed from its protective jacket. A disk can be made of hard plastic (hard disk) or soft plastic (floppy disk, diskette).

Diskette. See disk.

Disk drive. A machine that rotates a disk for the purpose of entering new information or copying existent information from the disk into the com-

puter's memory.

Display. Generally a television-style screen, it shows the user the information contained in the terminal memory.

Documentation. Written instructions for operating hardware or software.

DOS. Disk Operating System. (See Operating System.)

Dot matrix. The method by which most printers form alphanumeric characters or graphics, by printing a series of dots.

Double-density. Refers to the process of doubling the amount of information that can be placed on a disk or cassette.

Downtime. When a computer is "down," or unoperational.

Dual Intensity. Refers to a printer that can produce bold type in additional to ordinary type.

Editor. A program which makes it possible to enter text into a computer.

Electronic mail. Telephone transmission (via a modem) of information between computer users.

EPROM. Erasable Programmable Read Only Memory. A device such as a cartridge or disk that can have data added to it or erased.

Floppy disk. See disk.

FORTRAN. Formula Translator. Advanced programming language used for complex mathematical operations.

Emulator. A program-translating device that enables software designed for a particular computer to run on a different one.

Firmware. ROM-contained data, built into the machine or added via software, that cannot be changed.

Flow chart. Diagram outlining procedure for writing a program.

Format. (or Initialize) To get a disk ready to accept information.

Function key. A key on a computer that can be assigned to perform a particular function.

Full duplex. Communication mode capable of simultaneous transmission and reception of data.

Graphics. Non-textual CRT displays: charts, symbols, etc.

Graphics tablet. A device which will transfer whatever is drawn on it onto the computer monitor.

Half duplex. Communication mode capable of transmission of data and reception of data, but not at the same time.

Hard copy. Information, when it has been transferred from memory to a computer printout, is hard copy.

Hardware. Mechanical, non-software components of a computer system: computer, disk drive, printer, etc.; also, the components of the components, such as transistors, circuits, etc. Also the components of the components of the . . .just kidding.

Hard disk. Nonremovable storage device that is faster and has a far greater storage capacity than a floppy disk. It is generally hermetically sealed within the unit.

Hexadecimal. A base-16 numeration method widely used with computers. Numbers run from 0 to 9, then from A to F. Hexadecimal (or "hex") numbers are identified with the suffix H.

High-level language. An easily learned programming language that resembles human languages such as English. Examples. BASIC. COBOL, FORTRAN. Low level languages (such as Assembly) require that every command and function be separately programmed.

High resolution. High-quality graphics capability when applied to a video terminal. An individual software program may itself contain high-resolution graphics, meaning detailed or colorful graphics. A subjective term and therefore frequently abused (especially on New Years Eve).

Impact printer. A typewriter-style printer that presses characters through a ribbon.

Index hole. A hole punched in a floppy disk that is used by the disk drive to locate the beginning of sector 0 on the disk.

Instruction. A command contained in a program.

Interface. A connecting device which can be electronic or can be contained in software, for making a computer compatible with a peripheral device.

I/O. Input/Output. The system of pathways which channel information into a computer (usually through a keyboard, joystick, mouse, light pen) and out of the computer (usually onto a screen or printer).

K. See "Kilobyte"

Kilobyte. See "K". Seriously, a kilobyte (or 1K) is equal to 1,024 bytes (often rounded off to 1,000 bytes), and is the measure of the memory hardware or software can contain (8K, 64K, etc., etc.). Obviously, the higher the number the more capable the hardware/software.

Letter-quality. Refers to a printer that uses formed characters, such as those contained on a daisy wheel. Compare dot matrix printer.

Library. Collection of computer programs.

Light Pen. A device that allows a user to draw line figures or make menu selections directly on the screen. Proper hardware port and software required.

Load. Entry of information into the computer from an external storage, such as cassette player or disk drive.

LOGO. Programming language useful for graphics. Primarily employed in young children's education, each command is built logically on the previous command.

Loop. Programming statement used to order repetition of a task. A particularly useful one is called a "fruitful loop."

Machine language. Language used by the computer, comprised of binary numbers, into which the computer must translate programming languages.

Megabyte. Equal to 1,048,576 bytes (1048 kilobytes). Often abbreviated Mb.

Megahertz. Electrical frequency measure equalling one million cycles per second. Often abbreviated MHz.

Membrane. A flat computer keyboard that has touch-sensitive areas for each character rather than full-travel or stepped keys.

Memory. The word used to indicate any given computer's capacity to store information (10K memory etc.). There are different kinds of memory (ROM, RAM) and, within any computer, different memory locations (for sound, screen displays etc.).

Menu. A screen display of programs, on a particular disk or options in a program.

Microcomputer. An integrated small computer. It contains a microprocessor, memory, and interfaces for inputting and outputting information. Perfect example? Commodore 64.

Microprocessor. The brains of the computer, where mathematics and logical functions are performed. Also called the CPU.

Modem. Modulator/Demodulator. A device that changes information in analog form into digital form, and vice versa, for the purpose of transmitting computer information across a telephone line.

Monitor. Screen for displaying computer information.

Mouse. An input device, usually containing a selection button. The user slides the mouse on the

desktop beside the computer, and the cursor will duplicate the mouse's movement on the screen. Used to make menu selections and re-arrange information.

MP/M. Multiprogramming control Program for Microprocessors. A variety of the CP/M operating system that can be used by several users at a time.

Multi-user system. A system with a central pool of data or applications that can be accessed simultaneously be several users.

Number crunching. Refers to a computer's carrying out intricate or dense arithmetic or numerical functions.

Nybble. Half a byte, or four bits. (Or two shaves and two haircuts.)

Object code. A code in binary form, produced by an assembler or compiler program, A source code, by contrast, must be translated by an assembler or compiler before it can be executed by the CPU.

Octal. A base-8 numeration method often used with microcomputers. Numbers run from 0 to 7, and are identified with the suffix Q.

On-line. The state of being hooked up to an active computer, as in the case of a printer or disk drive that is connected and operational.

Operating system. The program(s) that supplies the computer system with its operating information, including start-up instructions at the time the system is turned on and supervisory instructions each time a new program is loaded.

Output. The path system that channels information out of a computer (usually to a screen or printer).

Overstriking. A method of producing boldface type by directing the printer to hit a character more than once.

Parallel. An input/output system that submits 8 bits of data at a time. An interface would have to be installed between a computer's serial port and a parallel printer, for example.

PASCAL. A more sophisticated programming language than BASIC, using less memory and producing faster programs. Named after Blaise Pascal.

PEEK. A programming command generally meaning: examine (specified location) and report the value that is represented there.

Peripheral. A hardware accessory to a computer, such as a printer or a modem.

Pixel. Picture element. A dot of light on a TV or computer screen, the smallest light fragment that the computer can address. Graphics with high resolution are generally composed of very small, and therefore numerous pixels.

Plotter. Machine for printing lines or graphs.

POKE. A programming command that is used to place a new value into a specified memory location.

Printer. Machine used to print computer information onto paper. See dot matrix, letter quality, daisy wheel.

Program. As a noun: a set of instructions given to a computer to enable it to perform a particular function. As a verb: to input such information into a computer.

Programming language. A language used in the composition of a computer program.

PROM. Programmable Read Only Memory. A permanent storage system for data that can be programmed both by the manufacturer and by the user.

Quad-density. Refers to a two-sided, double-density disk which is able to store four times the information of a one-sided, normal density disk.

RAM. Random Access Memory. Volatile storage system for data that can be changed—added to, subtracted from, rearranged—by the user. Information stored in RAM must be saved before the system is shut off or it will be lost.

Read/write. Signifies that information can be both read from and written into memory (RAM or permanent storage).

Resolution. A measure of the sharpness of a CRT (cathode ray tube) picture. Expressed in pixels, it can refer to either the number of scanning lines on the terminal or the number of addressable pixels on the screen.

Reverse video. The displaying of dark characters on a light background.

RF Modulator. Used to change computer video signals into radio frequency signals that can be picked up by a TV antenna and displayed.

RGB. Video signal composed of red, green, and blue that has much higher resolution and brighter colors than the standard composite color TV signals. A monitor with three separate electron guns, rather than the single gun used by the average color TV, is required.

ROM. Read Only Memory. Data that is built into a computer or software, and cannot be changed, such as the information that operates the computer immediately after start-up. In a computergame, for example, backgrounds which do not change or are not affected by gameplay are ROM. Spaceships and projectiles, which constantly are updated, are RAM.

Save. Recording information from the computer's memory onto tape or a disk.

Scrolling. Moving up and down (or from left to right) through the displayed information contained in the computer's memory, only a certain portion of which can be displayed on the terminal screen at one time.

Sector. A part of the track of a storage disk.

Serial. An input or output system that submits one bit of data at a time.

Serial port. A computer's input or output port through which data is transmitted in serial fashion. This is generally done, with home computers, through an RS232C serial interface port.

Single-sided. A disk with only one side that can store data.

Software. Programs to run on a computer on tape, disk, or cartridge. Everything from *Donkey Kong* to *Data Base.*

Source code. Program written in English-like words via an editor program which needs to be translated (with an assembler or compiler) into a language the computer understands.

Sprite. A high resolution programmable object. Any graphic figure is a sprite; sprites can be changed and animated through commands in BASIC

Syntax. Rules of grammatical usage governing programming language, as with English and other languages.

Terminal. The screen that displays computer information.

Text. The words and numbers displayed on the CRT screen.

Tractor feed. Mechanism that holds fan-fold paper in place and moves it through the printer, using sprockets that fit into the holes on both sides of the paper.

Volatile storage. A storage device, such as RAM, that loses the data it contains when power is cut off.

Winchester. A variety of hard disk that is sealed in a container.

Word processor. Can refer to the complete system, usually consisting of computer, display, memory storage, and printer, that is used to produce written documents; also refers to a software program that enables a computer to perform word processing functions.

Write. The process of transferring data from memory to permanent storage.

Write protection. Process that prevents writing to, or erasing from, a disk. With 8-inch disks, this is accomplished by removing a tab from the jacket; With 5¼ disks, by affixing the tab to the jacket.

Commodore Software–
The Best Game in Town.



...Take on the world, toughen up your trigger finger and fire away...

Commodore is the best computer value in town... at home, at school and at work... with our exciting, easy to use, inexpensive VIC 20 and C64 computers.

We're fast becoming the best game in town when it comes to entertainment for the whole family... and at affordable prices.

THE BEST ARCADE IN TOWN can be in your own home with our exciting, faithful reproductions of the best of Bally Midway arcade games. Our **Kickman,** (which just received a coveted "Electronic Games" award for an arcade translation) lets you steer the unicycle to catch the falling objects, as they fall quicker and quicker!!

Gorf, Lazarian, and Omega Race give you the best in classic space action against the one-eyed leviathon, the droids or the evil Empire.

In **The Wizard of Wor** you attempt to defeat the Wizard and the Warriors, fighting your way through to the end. With the new Commodore "MAGIC VOICE"... It talks back to you too!!

You commandeer the fleet at sea with our version of **Seawolf,** and become the master tactician as you battle "it out" with enemy fleet.

Clowns and **Blueprint** round out our arcade entertainment package to keep your fingers nimble and your mind in gear.

commodore
COMPUTERS

First In Quality Software

See your local dealer now... He's got the best game in town... just for you.

Reader Service No. 89